

PATENT COOPERATION TREATY

From the INTERNATIONAL BUREAU

PCT

NOTIFICATION OF ELECTION

(PCT Rule 61.2)

To:

Commissioner **US Department of Commerce** United States Patent and Trademark Office, PCT 2011 South Clark Place Room CP2/5C24 Arlington, VA 22202

ETATS-UNIS D'AMERIQUE Date of mailing (day/month/year) 10 November 2000 (10.11.00)

in its capacity as elected Office

International application No.	
PCT/EP00/02070	
International filing date (day/mon	th/y

Applicant

Applicant's or agent's file reference 99P1391P

Priority date (day/month/year) ear) 10 March 1999 (10.03.99) 09 March 2000 (09.03.00)

NEUHAUS	, Matthias et al	

1.	The designated Office is hereby notified of its election made:
	X in the demand filed with the International Preliminary Examining Authority on:
	09 October 2000 (09.10.00)
	in a notice effecting later election filed with the International Bureau on:
2.	The election X was
	made before the expiration of 19 months from the priority date or, where Rule 32 applies, within the time limit under Rule 32.2(b).

The International Bureau of WIPO 34, chemin des Colombettes 1211 Geneva 20, Switzerland

Authorized officer

Zakaria EL KHODARY

Telephone No.: (41-22) 338.83.38

Facsimile No.: (41-22) 740.14.35

PATENT COOPERSION TREATY PCT INTERNATIONAL PRELIMINARY EXAMINATION REPORT

(PCT Article 36 and Rule 70)

Applicant's or agent's file reference 99P1391P	FOR FURTHER ACTION	FOR FURTHER ACTION SeeNotificationofTransmittalofInternational Prelim Examination Report (Form PCT/IPEA/416)							
International application No. PCT/EP00/02070	International filing date (day/n 09 March 2000 (09.0		Priority date (day/month/year) 10 March 1999 (10.03.99)						
International Patent Classification (IPC) or national classification and IPC H04Q 3/00									
Applicant	SIEMENS AKTIENGESE	LLSCHAFT							
and is transmitted to the applicant ac	ccording to Article 36.		ational Preliminary Examining Authority						
2. This REPORT consists of a total of	6 sheets, including	ng this cover sh	neet.						
amended and are the basis fo	ied by ANNEXES, i.e., sheets or this report and/or sheets conta Administrative Instructions und	ining rectificat	on, claims and/or drawings which have been tions made before this Authority (see Rule						
These annexes consist of a to	otal of sheets.								
3. This report contains indications rela	ating to the following items:								
I Basis of the report									
II Priority									
III Non-establishment	of opinion with regard to novelt	y, inventive ste	ep and industrial applicability						
IV Lack of unity of inv	vention								
Reasoned statemen		l to novelty, in it	ventive step or industrial applicability;						
VI Certain documents	cited								
VII Certain defects in the	he international application								
	ns on the international application	n							
Date of submission of the demand	Date of	of completion of	of this report						
09 October 2000 (09.	10.00)	08	June 2001 (08.06.2001)						
Name and mailing address of the IPEA/EP	Autho	Authorized officer							
Facsimile No.	Telep	hone No.							





INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No.

PCT/EP00/02070

I. Basis of the report									
1.	With	regard to	the eleme	nts of the in	ternational ap	plication:*			
		the inte	rnational ap	oplication as	originally fil	led			
	$\overline{\boxtimes}$	the desc	cription:						
		pages				1,3-23		, as originally filed	
		pages						, filed with the demand	
		pages			2,2a		_, filed with the letter of	09 March 2001 (09.03.2001)	
	∇	the clai	me.						
		pages						, as originally filed	
		pages					1 1 6 4	ner with any statement under Article 19	
		pages		- 44				, filed with the demand	
		pages			1-17		_, filed with the letter of	09 March 2001 (09.03.2001)	
			•						
		the drav	_			1/5-5/5		, as originally filed	
		pages						, filed with the demand	
		pages					filed with the letter of		
		pages					_, med with the letter of		
	U	he seque	_	part of the c					
		pages						, as originally filed	
		pages						, filed with the demand	
		pages					_, filed with the letter of		
2.	the in	nternation the land the land	mal applicants were avanguage of an applicant application and applications.	tion was file ailable or fun translation f ublication or	ed, unless other nished to this furnished for f the internati	erwise indicated is Authority in the the purposes of it onal application	under this item. e following language nternational search (under (under Rule 48.3(b)).		
3.		or 55.3	3).					ary examination (under Rule 55.2 and mational application, the internationa	
Э.	preli	minary e	examination	n was carried	d out on the b	asis of the seque	nce listing:		
	H					n written form.	e eardable form		
	H	filed together with the international application in computer readable form.							
	H	furnished subsequently to this Authority in written form.							
		furnished subsequently to this Authority in computer readable form. The statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.							
			tatement th	at the info	rmation reco	rded in compute	r readable form is identic	cal to the written sequence listing ha	
4.		The au	mendments	have resulte	ed in the cand	cellation of:			
			the descri	ption, pages					
		\Box							
l					ig				
5.	. 🔲	This re	eport has be	en establish	ned as if (son	ne of) the amend	ments had not been made ental Box (Rule 70.2(c)).**	e, since they have been considered to g	
•	in th	lacement his repoi 70.17).	sheets whi rt as "orig	ch have bee ginally filed	en furnished t '" and are n	to the receiving (not annexed to	Office in response to an im his report since they do	vitation under Article 14 are referred to not contain amendments (Rule 70.1	
١			nent sheet (containing s	uch amendme	ents must be refe	red to under item 1 and a	nnexed to this report.	

International application No. PCT/EP 00/02070

v.	Reasoned statement under Article 3 citations and explanations supportin		velty, inventive step or industrial appli	cability;
1.	Statement			
	Novelty (N)	Claims	1-17	YES
		Claims		NO
	Inventive step (IS)	Claims	1-17	YES
		Claims		NO
	Industrial applicability (IA)	Claims	1-17	YES
		Claims		NO

2. Citations and explanations

Citations

1. This <u>international preliminary examination report</u> makes reference to the following document:

D1: EP-A-0 817 422.

- The present international application relates to a "method for operating a telecommunications network" as per the preamble to Claim 1, a "network element for operating a telecommunications network" as per the preamble to Claim 16, and a corresponding "telecommunications network" as per Claim 17.
- 1.1 Basis of and background to the present invention:

 A network element of the telecommunications network has a control computer with a plurality of application programs which, when implemented, operate application objects. A connection is established between an external operating computer and said control computer, the control computer being serviced via said connection.

If the application programmes are developed further

in the control computer, then it must be guaranteed that application objects which are regarded by the operating computer as belonging to a certain <u>class</u> can also be processed by said operating computer if such objects were assigned to a different <u>class</u>, i.e. a <u>substitute class</u>, in the developed application programme.

1.2 Prior art:

(a) CCITT X.720 (01/92), section 5.2.3, "allomorphism"

The cited standard proposes a programming technique for application programmes that takes <u>allomorphism</u> into account, i.e. a certain application object in the application programme can be implemented by the operating computer as if it were an object belonging to the class known in the operating computer. The <u>class</u> known in the operating computer is <u>allomorphic</u> to the class actually known in the application programme, i.e. the substitute class.

- (b) Document **D1** also describes a method for "managing" objects. In **D1**, a given first object can also communicate with future unknown second objects. For this purpose, when the first object is established, a so-called "abstract object" with corresponding interfaces is devised. If a second object unknown to the first object is later established, this second object inherits the properties and interfaces of the abstract object.
- The present <u>international application</u> addresses the <u>technical problem</u> of developing a "<u>method for operating a telecommunications network"</u> such that the application programmes are largely free of

matching with regard to the association between substitute classes and application objects. This should result in an efficient method for sending messages from an operating computer to the objects of application programmes in the control computer of a network element.

is solved by the features of the characterising part of Claim 1, according to which an interface programme is created in the control computer of a network element, said programme, when it encounters a message sent from an operating computer to an application programme in said control computer, converting the class designation of an object contained in the message and known to the operating computer into a substitute designation that provides a substitute class for the object in the application programme. Thus, when a message is processed in the application programme, the object is regarded as belonging to the substitute class.

The claimed method is advantageous in that the operation that assigns <u>substitute designations</u> to <u>class designations</u> does not have to be programmed in each application programme, but only once at the central point in the <u>interface programme</u>.

- 4. The claimed definition of the "method", as specified in the features of the characterising part of Claim 1, is neither disclosed nor suggested by the prior art:
- 4.1 The aforementioned X.720 standard does not include indications on implementation that would lead a

person skilled in the art to a solution as per Claim 1, using an interface programme to convert a class into a substitute class.

- 4.2 Document D1 contains no suggestion of the <u>allocation</u> of substitute classes, nor the implementation of an <u>interface programme</u> before the application programme for converting <u>class designations</u> into <u>substitute</u> designations.
 - Claim 1 therefore meets the requirements for novelty and inventive step of PCT Article 33(2) and (3).
- Dependent Claims 2-15 are all directly or indirectly dependent on Claim 1 and therefore likewise meet the requirements for novelty and inventive step of PCT Article 33(2) and (3).
- 6. Claim 16, which relates to a "network element for operating a telecommunications network" also contains, by way of reference to the preceding method claims (i.e. by the feature "and that the control computer, by executing the programme, implements the method according to one of the preceding claims"), the inventive features of independent Claim 1 and therefore likewise meets the requirements for novelty and inventive step of PCT Article 33(2) and (3).
- 7. Claim 17, which relates to a "telecommunications network", is dependent on Claim 16 and therefore likewise meets the requirements for novelty and inventive step of PCT Article 33(2) and (3).

VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS

Absender:

MIT DER INTERNATIONALEN VORLÄUFIGEN PRÜFUNG BEAUFTRAGTE BEHÖRDE

An:

SIEMENS AKTIENGESELLSCHAFT

Postfach 22 16 34

D-80506 München ALLEMAGNE

Eing. 11. Juni 2001 Jom GR 10.07.2001 PCT

MITTEILUNG ÜBER DIE ÜBERSENDUNG DES INTERNATIONALEN VORLÄUFIGEN PRÜFUNGSBERICHTS

(Regel 71.1 PCT)

Absendedatum

(Tag/Monat/Jahr)

08.06.2001

Aktenzeichen des Anmelders oder Anwalts

99P1391P

WICHTIGE MITTEILUNG

Internationales Aktenzeichen PCT/EP00/02070

Internationales Anmeldedatum (Tag/Monat/Jahr) 09/03/2000

Prioritätsdatum (Tag/Monat/Jahr)

10/03/1999

Anmelder

SIEMENS AKTIENGESELLSCHAFT et al.



- 1. Dem Anmelder wird mitgeteilt, daß ihm die mit der internationalen vorläufigen Prüfung beauftragte Behörde hiermit den zu der internationalen Anmeldung erstellten internationalen vorläufigen Prüfungsbericht, gegebenenfalls mit den dazugehörigen Anlagen, übermittelt.
- 2. Eine Kopie des Berichts wird gegebenenfalls mit den dazugehörigen Anlagen dem Internationalen Büro zur Weiterleitung an alle ausgewählten Ämter übermittelt.
- 3. Auf Wunsch eines ausgewählten Amts wird das Internationale Büro eine Übersetzung des Berichts (jedoch nicht der Anlagen) ins Englische anfertigen und diesem Amt übermitteln.

4. ERINNERUNG



Zum Eintritt in die nationale Phase hat der Anmelder vor jedem ausgewählten Amt innerhalb von 30 Monaten ab dem Prioritätsdatum (oder in manchen Ämtern noch später) bestimmte Handlungen (Einreichung von Übersetzungen und Entrichtung nationaler Gebühren) vorzunehmen (Artikel 39 (1)) (siehe auch die durch das Internationale Büro im Formblatt PCT/IB/301 übermittelte Information).

lst einem ausgewählten Amt eine Übersetzung der internationalen Anmeldung zu übermitteln, so muß diese Übersetzung auch Übersetzungen aller Anlagen zum internationalen vorläufigen Prüfungsbericht enthalten. Es ist Aufgabe des Anmelders, solche Übersetzungen anzufertigen und den betroffenen ausgewählten Ämtern direkt zuzuleiten.

Weitere Einzelheiten zu den maßgebenden Fristen und Erfordernissen der ausgewählten Ämter sind Band II des PCT-Leitfadens für Anmelder zu entnehmen.

Name und Postanschrift der mit der internationalen Prüfung beauftragten Behörde

Europäisches Patentamt D-80298 München

Tel. +49 89 2399 - 0 Tx: 523656 epmu d

Fax: +49 89 2399 - 4465

Bevollmächtigter Bediensteter

Finnie, A

Tel. +49 89 2399-8251



VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS

Ábsender: INTERNATIONALE RECHERCHENBEHÖRDE

PCT

SIEMENS AKTIENGESELLSCHAFT Postfach 22 16 34 D-80506 München **GERMANY** ZT GG VM : Cr Eing. 23. Mai 2000 GR Frist

MITTEILUNG ÜBER DIE ÜBERMITTLUNG DES INTERNATIONALEN RECHERCHENBERICHTS ODER DER ERKLÄRUNG

(Regel 44.1 PCT)

Absendedatum

(Tag/Monat/Jahr)

19/05/2000

09/03/2000

Aktenzeichen des Anmelders oder Anwalts

99P1391P

WEITERES VORGEHEN

siehe Punkte 1 und 4 unten

Internationales Aktenzeichen

PCT/EP 00/02070

Internationales Anmeldedatum (Tag/Monat/Jahr)

Anmelder

SIEMENS AKTIENGESELLSCHAFT

1. X Dem Anmelder wird mitgeteilt, daß der internationale Recherchenbericht erstellt wurde und ihm hiermit übermittelt wird.

Einreichung von Änderungen und einer Erklärung nach Artikel 19:

Der Anmelder kann auf eigenen Wunsch die Ansprüche der internationalen Anmeldung ändern (siehe Regel 46);

Bis wann sind Anderungen einzureichen?

Die Frist zur Einreichung solcher Änderungen beträgt üblicherweise zwei Monate ab der Übermittlung des internationalen Recherchenberichts; weitere Einzelheiten sind den Anmerkungen auf dem Beiblatt zu entnehmen.

Wo sind Anderungen einzureichen?

Unmittelbar beim Internationalen Büro der WIPO, 34, CHEMIN des Colombettes, CH-1211 Genf 20,

Telefaxnr.: (41-22) 740.14.35

Nähere Hinwelse sind den Anmerkungen auf dem Beiblatt zu entnehmen.

Dem Anmelder wird mitgeteilt, daß kein internationaler Recherchenbericht erstellt wird und daß ihm hiermit die Erklärung nach Artikel 17(2)a) übermittelt wird.

Hinsichtlich des Widerspruchs gegen die Entrichtung einer zusätzlichen Gebühr (zusätzlicher Gebühren) nach Regel 40.2 wird dem Anmelder mitgeteilt, daß

der Widerspruch und die Entscheidung hierüber zusammen mit seinem Antrag auf Übermittlung des Wortlauts sowohl des Widerspruchs als auch der Entscheidung hierüber an die Bestimmungsämter dem Internationalen Buro übermittelt worden

noch keine Entscheidung über den Widerspruch vorliegt; der Anmelder wird benachrichtigt, sobald eine Entscheidung getroffen wurde.

4. Welteres Vorgehen: Der Anmelder wird auf folgendes aufmerksam gemacht:

Weiteres vorgenen:

Der Allineider wird das regenende anneldung vom Internationalen Büre veröffentKurz nach Ablauf von 18 Monaten seit dem Prioritätsdatum wird die internationale Anmeldung vom Internationalen Büre veröffentlicht. Will der Anmelder die Veröffentlichung verhindem oder auf einen späteren Zeitpunkt verschieben, so muß gemäß Regel 90 by. bzw. 90 3 vor Abschluß der technischen Vorbereitungen für die internationale Veröffentlichung eine Erklärung über die Zurücknahme der internationalen Anmeldung oder des Prioritätsanspruchs beim Internationalen Büro eingehen.

Innerhalb von 19 Monaten seit dem Prioritätsdatum ist ein Antrag auf internationale vorläufige Prüfung einzureichen, wenn der Anmelder den Eintritt in die nationale Phase bis zu 30 Monaten seit dem Prioritätsdatum (in manchen Ämtern sogar noch länger) verschiehen möchte

Innerhalb von 20 Monaten seit dem Prioritätsdatum muß der Anmelder die für den Eintritt in die nationale Phase vorgeschriebenen Handlungen vor allen Bestimmungsamtern vornehmen, die nicht innerhalb von 19 Monaten seit dem Prioritätsdatum in der Anmeldung oder einer nachträglichen Auswahlerklärung ausgewählt wurden oder nicht ausgewählt werden konnten, da für sie Kapitel II des Vertrages nicht verbindlich ist.

Name und Postanschrift der Internationalen Recherchenbehörde

Europäisches Patentamt, P.B. 5818 Patentlaan 2 NL-2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo ni,

Fax: (+31-70) 340-3016

Bevollmächtigter Bediensteter

Theresia Van Deursen



ANMERKUNGEN ZU FORMBLATT PCT/ISA/220

Diese Anmerkungen sollen grundlegende Hinweise zur Einreichung von Änderungen gemäß Artikel 19 geben. Diesen Anmerkungen liegen die Erfordernisse des Vertrags über die internationale Zusammenarbeit auf dem Gebiet des Patentwesens (PCT), der Ausführungsordnung und der Verwaltungsrichtlinien zu diesem Vertrag zugrunde. Bei Abweichungen zwischen diesen Anmerkungen und
obengenannten Texten sind letztere maßgebend. Nähere Einzelheiten sind dem PCT-Leitfaden für Anmelder, einer Veröffentlichung der
WIPO, zu entnehmen.

Die in diesen Anmerkungen verwendeten Begriffe "Artikel", "Regel" und "Abschnitt" beziehen sich jeweils auf die Bestimmungen des PCT-Vertrags, der PCT-Ausführungsordnung bzw. der PCT-Verwaltungsrichtlinien.

HINWEISE ZU ÄNDERUNGEN GEMÄSS ARTIKEL 19

Nach Erhalt des internationalen Recherchenberichts hat der Anmelder die Möglichkeit, einmal die Ansprüche der internationalen Anmeldung zu ändern. Es ist jedoch zu betonen, daß, da alle Teile der internationalen Anmeldung (Ansprüche, Beschreibung und Zeichnungen) während des internationalen vorläufigen Prüfungsverfahrens geändert werden können, normalerweise keine Notwendigkeit besteht, Änderungen der Ansprüche nach Artikel 19 einzureichen, außer wenn der Anmelder z.B. zum Zwecke eines vorläufigen Schutzes die Veröffentlichung dieser Ansprüche wünscht oder ein anderer Grund für eine Änderung der Ansprüche vor ihrer internationalen Veröffentlichung vorliegt. Weiterhin ist zu beachten, daß ein vorläufiger Schutz nur in einigen Staaten erhältlich ist.

Welche Teile der internationalen Anmeldung können geändert werden?

Im Rahmen von Artikel 19 können nur die Ansprüche geändert werden.

In der internationalen Phase können die Ansprüche auch nach Artikel 34 vor der mit der internationalen vorläufigen Prüfung beauftragten Behörde geändert (oder nochmals geändert) werden. Die Beschreibung und die Zeichnungen können nur nach Artikel 34 vor der mit der internationalen vorläufigen Prüfung beauftragten Behörde geändert werden.

Beim Eintritt in die nationale Phase können alle Teile der internationalen Anmeldung nach Artikel 28 oder gegebenenfalls Artikel 41 geändert werden.

Bis wann sind Änderungen einzureichen?

Innerhalb von zwei Monaten ab der Übermittlung des internationalen Recherchenberichts oder innerhalb von sechzehn Monaten ab dem Prioritätsdatum, je nachdem, welche Frist später abläuft. Die Änderungen gelten jedoch als rechtzeitig eingereicht, wenn sie dem Internationalen Büro nach Ablauf der maßgebenden Frist, aber noch vor Abschluß der technischen Vorbereitungen für die internationale Veröffentlichung (Regel 46.1) zugehen.

Wo sind die Änderungen nicht einzureichen?

Die Änderungen können nur beim Internationalen Büro, nicht aber beim Anmeldeamt oder der Internationalen Recherchenbehörde eingereicht werden (Regel 46.2).

Falls ein Antrag auf internationale vorläufige Prüfung eingereicht wurde/wird, siehe unten.

In welcher Form können Änderungen erfolgen?

Eine Änderung kann erfolgen durch Streichung eines oder mehrerer ganzer Ansprüche, durch Hinzufügung eines oder mehrerer neuer Ansprüche oder durch Änderung des Wortlauts eines oder mehrerer Ansprüche in der eingereichten Fassung.

Für jedes Anspruchsblatt, das sich aufgrund einer oder mehrerer Änderungen von dem ursprünglich eingereichten Blatt unterscheidet, ist ein Ersatzblatt einzureichen.

Alle Ansprüche, die auf einem Ersatzblatt erscheinen, sind mit arabischen Ziffern zu numerieren. Wird ein Ansprüch gestrichen, so brauchen, die anderen Ansprüche nicht neu numeriert zu werden. Im Fall einer Neunumerierung sind die Ansprüche fortlaufend zu numerieren (Verwaltungsrichtlinien, Abschnitt 205 b)).

Die Änderungen sind in der Sprache abzufassen, in der dieinternationale Anmeidung veröffentlicht wird.

Welche Unterlagen sind den Änderungen beizufügen?

Begleitschreiben (Abschnitt 205 b)):

Die Änderungen sind mit einem Begleitschreiben einzureichen.

Das Begleitschreiben wird nicht zusammen mit der internationalen Anmeldung und den geänderten Ansprüchen veröffentlicht. Es ist nicht zu verwechseln mit der "Erklärung nach Artikel 19(1)" (siehe unten, "Erklärung nach Artikel 19 (1)").

Das Begleitschreiben ist nach Wahl des Anmelders in englischer oder französischer Sprache abzufassen. Bei englischsprachigen internationalen Anmeldungen ist das Begleitschreiben aber ebenfalls in englischer, bei französischsprachigen internationalen Anmeldungen in französischer Sprache abzufassen.

ANMERKUNGEN ZU FORMBLATT PCT/ISA/220 (Fortsetzung)

Im Begleitschreiben sind die Unterschiede zwischen den Ansprüchen in der eingereichten Fassung und den geänderten Ansprüchen anzugeben. So ist insbesondere zu jedem Ansprüch in der internationalen Anmeldung anzugeben (gleichlautende Angaben zu verschiedenen Ansprüchen können zusammengefaßt werden), ob

- i) der Anspruch unverändert ist;
- ii) der Anspruch gestrichen worden ist;
- iii) der Anspruch neu ist;
- iv) der Anspruch einen oder mehrere Ansprüche in der eingereichten Fassung ersetzt;
- v) der Anspruch auf die Teilung eines Anspruchs in der eingereichten Fassung zurückzuführen ist.

Im folgenden sind Belspiele angegeben, wie Änderungen im Begleitschreiben zu erläutern sind:

- [Wenn anstelle von ursprünglich 48 Ansprüchen nach der Änderung einiger Ansprüche 51 Ansprüche existieren]:
 "Die Ansprüche 1 bis 29, 31, 32, 34, 35, 37 bis 48 werden durch geänderte Ansprüche gleicher Numerierung ersetzt; Ansprüche 30, 33 und 36 unverändert; neue Ansprüche 49 bis 51 hinzugefügt."
- (Wenn anstelle von ursprünglich 15 Ansprüchen nach der Änderung aller Ansprüche 11 Ansprüche existieren): "Geänderte Ansprüche 1 bis 11 treten an die Stelle der Ansprüche 1 bis 15."
- 3. [Wenn ursprünglich 14 Ansprüche existierten und die Änderungen darin bestehen, daß einige Ansprüche gestrichen werden und neue Ansprüche hinzugefügt werden]: Ansprüche 1 bis 6 und 14 unverändert; Ansprüche 7 bis 13 gestrichen; neue Ansprüche 15, 16 und 17 hinzugefügt. "Oder" Ansprüche 7 bis 13 gestrichen; neue Ansprüche 15, 16 und 17 hinzugefügt; alle übrigen Ansprüche unverändert."
- [Wenn verschiedene Arten von Änderungen durchgeführt werden]:
 "Ansprüche 1-10 unverändert; Ansprüche 11 bis 13, 18 und 19 gestrichen; Ansprüche 14, 15 und 16 durch geänderten Ansprüch 14 ersetzt; Ansprüch 17 in geänderte Ansprüche 15, 16 und 17 unterteilt; neue Ansprüche 20 und 21 hinzugefügt."

"Erklärung nach Artikel 19(1)" (Regel 46.4)

Den Änderungen kann eine Erklärung beigefügt werden, mit der die Änderungen erläutert und ihre Auswirkungen auf die Beschreibung und die Zeichnungen dargelegt werden (die nicht nach Artikel 19 (1) geändert werden können).

Die Erklärung wird zusammen mit der internationalen Anmeldung und den geänderten Ansprüchen veröffentlicht.

Sie ist in der Sprache abzufassen, in der die internationalen Anmeldung veröffentlicht wird.

Sie muß kurz gehalten sein und darf, wenn in englischer Sprache abgefaßt oder ins Englische übersetzt, nicht mehr als 500 Wörter umfassen

Die Erklärung ist nicht zu verwechseln mit dem Begleitschreiben, das auf die Unterschiede zwischen den Ansprüchen in der eingereichten Fassung und den geänderten Ansprüchen hinweist, und ersetzt letzteres nicht. Sie ist auf einem gesonderten Blatt einzureichen und in der Überschrift als solche zu kennzeichnen, vorzugsweise mit den Worten "Erklärung nach Artikel 19 (1)".

Die Erklärung darf keine herabsetzenden Äußerungen über den inter nationalen Recherchenbericht oder die Bedeutung von in dem Bericht angeführten Veröffentlichungen enthalten. Sie darf auf im internationalen Recherchenbericht angeführte Veröffentlichungen, die sich auf einen bestimmten Anspruch beziehen, nur im Zusammenhang mit einer Änderung dieses Anspruchs Bezug nehmen.

Auswirkungen eines bereits gestellten Antrags auf internationalevorläufige Prüfung

lst zum Zeitpunkt der Einreichung von Änderungen nach Artikel 19 bereits ein Antrag auf internationale vorläufige Prüfung gestellt worden, so sollte der Anmelder in seinem Interesse gleichzeitig mit der Einreichung der Änderungen beim Internation alen Büro auch eine Kopie der Änderungen bei der mit der internationalen vorläufigen Prüfung beauftragen Behörde einreichen (siehe Regel 62.2 a), erster Satz).

Auswirkungen von Änderungen hinsichtlich der Übersetzung derinternationalen Anmeldung beim Eintritt in die nationale Phase

Der Anmelder wird darauf hingewiesen, daß bei Eintritt in die nationale Phase möglicherweise anstatt oder zusätzlich zu der Übersetzung der Ansprüche in der eingereichten Fassung eine Übersetzung der nach Artikel 19 geänderten Ansprüche an die bestimmten/ausgewählten Ämter zu übermitteln ist.

Nähere Einzelheiten über die Erfordemisse jedes bestimmten/ausgewählten Amts sind Band II des PCT-Leitfadens für Anmelder zu entnehmen.

PCT

INTERNATIONALE ZUSAMENARBEIT AUF DEM

PCT

(Artikel 36 und Regel 70 PCT)

		A Ida ada Aassalla	(, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		
99P1391F		Anmelders oder Anwalts	WEITERES VORGE	siehe Mittei HEN vorläufigen	ilung über die Übersendung des internationalen Prüfungsberichts (Formblatt PCT/IPEA/416)
International	es Ak	tenzeichen	Internationales Anmeldeda	atum <i>(Tag/Monat/Jahr)</i>	Prioritätsdatum (Tag/Monat/Tag)
PCT/EP00			09/03/2000	, ,	10/03/1999
	e Pate		nationale Klassifikation und	IPK	
					,
Anmelder					
SIEMENS	AK'	TIENGESELLSCHAFT	et al.		
			fungsbericht wurde von d elder gemäß Artikel 36 ü		onalen vorläufigen Prüfung beauftragten
2. Dieser	BEF	IICHT umfaßt insgesamt	6 Blätter einschließlich	dieses Deckblatts.	
un B€	d/od hörd	er Zeichnungen, die geä	ndert wurden und dieser chtigungen (siehe Regel	m Bericht zugrunde	åtter mit Beschreibungen, Ansprüchen liegen, und/oder Blätter mit vor dieser itt 607 der Verwaltungsrichtlinien zum PCT).
Diese	Ailia	gen umassen msgesam	it / Blatter.		
	_	cht enthält Angaben zu f			
	⊠ □	Grundlage des Berichts	3		
		Priorität	Outrabtone über Naubo	it orfindoriocho Töt	iakait und anwarhlicha Anwandharkait
		-		ii, eminuensche Tai	igkeit und gewerbliche Anwendbarkeit
V V	Ճ	Begründete Feststellun	g nach Artikel 35(2) hins	sichtlich der Neuheit Erklärungen zur Stüt	t, der erfinderischen Tätigkeit und der tzung dieser Feststellung
l vi		Bestimmte angeführte	Unterlagen		
VII		Bestimmte Mängel der	internationalen Anmeldu	ıng	
VIII		Bestimmte Bemerkung	en zur internationalen Aı	nmeldung	
Datum der E	inreid	chung des Antrags		Datum der Fertigstell	ung dieses Berichts
09/10/200	09/10/2000				
	uftraç	nschrift der mit der internation gten Behörde:	onalen vorläufigen	Bevollmächtigter Bed	liensteter
	D-80	päisches Patentamt 0298 München +49 89 2399 - 0 Tx: 52365	6 epmu d	Möll, H-P	
		+49 89 2399 - 4465		Tel. Nr. +49 89 2399	8243



Internationales Aktenzeichen PCT/EP00/02070

I. Gr	undlage	des Be	ric	hts
-------	---------	--------	-----	-----

۱.	Auff eing	insichtlich der Bestandteile der internationalen Anmeldung (<i>Ersatzblätter, die dem Anmeldeamt auf eine</i> ufforderung nach Artikel 14 hin vorgelegt wurden, gelten im Rahmen dieses Berichts als "ursprünglich ngereicht" und sind ihm nicht beigefügt, weil sie keine Änderungen enthalten (Regeln 70.16 und 70.17)): es chreibung, Seiten:									
	1,3-	23	ursprüngliche Fassung								
	2,2a	ı	eingegangen am	12/03/2001	mit Schreiben vom	09/03/2001					
	Patentansprüche, Nr.:										
	1-17	7	eingegangen am	12/03/2001	mit Schreiben vom	09/03/2001					
	Zeid	chnungen, Blätter	:								
	1/5-	5/5	ursprüngliche Fassung								
2.	die i	nternationale Anm	he: Alle vorstehend genannten E eldung eingereicht worden ist, z chts anderes angegeben ist.								
		Bestandteile stand ereicht; dabei hand	en der Behörde in der Sprache: delt es sich um	zur Verfügu	ng bzw. wurden in die	ser Sprache					
		die Sprache der Ü Regel 23.1(b)).	bersetzung, die für die Zwecke	der internatio	nalen Recherche eing	ereicht worden ist (nach					
		die Veröffentlichu	ngssprache der internationalen A	Anmeldung (n	ach Regel 48.3(b)).						
			bersetzung, die für die Zwecke 5.2 und/oder 55.3).	der internatio	nalen vorläufigen Prüf	ung eingereicht worden					
3.			internationalen Anmeldung offer je Prüfung auf der Grundlage de								
		in der internationa	len Anmeldung in schriftlicher F	orm enthalten	ist.						
		zusammen mit de	r internationalen Anmeldung in d	computerlesba	arer Form eingereicht	worden ist.					
		bei der Behörde n	achträglich in schriftlicher Form	eingereicht w	orden ist.						
		bei der Behörde n	achträglich in computerlesbarer	Form eingere	eicht worden ist.						
			ß das nachträglich eingereichte alt der internationalen Anmeldun								
			ß die in computerlesbarer Form entsprechen, wurde vorgelegt.	erfassten Info	rmationen dem schrift	tlichen					





Internationales Aktenzeichen PCT/EP00/02070

4.	Auf	grund der Änderunge	n sind folgende U	Interlagen fort	gefallen:			
		Beschreibung,	Seiten:					
		Ansprüche,	Nr.:					
		Zeichnungen,	Blatt:					
5.		Dieser Bericht ist oh angegebenen Gründ eingereichten Fassu	len nach Auffassi	ung der Behör	de über den	•		
		(Auf Ersatzblätter, di beizufügen).	ie solche Änderui	ngen enthalter	n, ist unter Pu	unkt 1 hinzuwei	sen;sie sind dies	sem Bericht
6.	Etw	aige zusätzliche Bem	erkungen:					
V.		gründete Feststellun verblichen Anwendb						keit und de
1.	Fes	tstellung						
	Neu	uheit (N)	Ja: Nein	Ansprüche : Ansprüche	1-17			
	Erfi	nderische Tätigkeit (E	•	Ansprüche : Ansprüche	1-17			
	Gev	verbliche Anwendbarl		Ansprüche : Ansprüche	1-17			
2.	Unt	erlagen und Erklärung	gen					

siehe Beiblatt



Angeführte Unterlagen

In diesem Internationalen Vorläufigen Prüfungsbericht wird auf das folgende 1. Dokument verwiesen:

D1: EP - A - 0 817 422

Zu Punkt V

Begründete Feststellung nach Artikel 35(2) hinsichtlich der Neuheit, der erfinderischen Tätigkeit und der gewerblichen Anwendbarkeit; Unterlagen und Erklärungen zur Stützung dieser Feststellung

- Die vorliegende Internationale Anmeldung betrifft ein "Verfahren zum Betreiben 1. eines Telekommunikationsnetzes" gemäß Oberbegriff des Anspruchs 1, ein "Netzelement zum Betreiben eines Telekommunikationsnetzes" gemäß Oberbegriff des Anspruchs 16 sowie ein entsprechendes "Telekommunikationsnetz" gemäß Anspruch 17.
- 1.1 Grundlage und Hintergrund der vorliegenden Erfindung:

Ein Netzelement des Telekommunikationsnetzes verfügt über einen Steuerrechner mit mehreren Anwendungsprogrammen, bei deren Ausführen Anwendungsobjekte bearbeitet werden. Zwischen einem externen Bedienrechner und diesem Steuerrechner wird eine Verbindung aufgebaut über die der Steuerrechner gewartet wird.

Werden nun die Anwendungsprogramme im Steuerrechner weiterentwickelt, so muß gewährleistet sein, daß Anwendungsobjekte, die vom Bedienrechner als zu einer bestimmten Klasse zugehörig angesehen werden, auch dann von diesem Bedienrechner aus behandelt werden können, wenn diese Objekte im weiterentwickelten Anwendungsprogramm einer geänderten Klasse, d.h. einer Ersatzklasse, zugeordnet wurden.

- 1.2 Stand der Technik:
- (a) CCITT X.720 (01/92), Abschnitt 5.2.3, "Allomorphie" Im genannten Standard wird für die Anwendungsprogramme eine Programmiertechnik vorgeschlagen, die Allomorphie berücksichtigt, d.h. ein

INTERNATIONALER VORLÄUFIGER PRÜFUNGSBERICHT - BEIBLATT



bestimmtes Anwendungsobjekt im Anwendungsprogramm kann vom Bedienrechner aus so geführt werden, als wäre es ein Objekt der im Bedienrechner bekannten Klasse. Die im Bedienrechner bekannte Klasse ist zur tatsächlich im Anwendungsprogramm bekannten Klasse, d.h. der Ersatzklasse, allomorph.

- Das Dokument D1 beschreibt ebenfalls ein Verfahren zum "Managen" von (b) Objekten. In D1 wird erreicht, daß ein gegebenes erstes Objekt auch mit unbekannten zukünftigen zweiten Objekten kommunizieren kann. Dazu wird bei Einrichtung des ersten Objekts ein sogenanntes abstraktes Objekt ("abstract object") mit entsprechenden Schnittstellen geschaffen. Wird nun später ein zweites Objekt, welches dem ersten Objekt unbekannt ist eingerichtet, so erbt dieses zweite Objekt die Eigenschaften und Schnittstellen des abstrakten Objekts.
- Es ist die technische Aufgabe der vorliegenden Internationalen Anmeldung, ein 2. "Verfahren zum Betreiben eines Telekommunikationsnetzes" derart bereitzustellen, daß die Anwendungprogramme weitestgehend von Anpassungsaufgaben hinsichtlich der Zuordnung von Ersatzklassen zu Anwendungsobjekten entlastet werden. Somit soll ein effizientes Verfahren zum Senden von Nachrichten von einem Bedienrechner an die Objekte von Anwendungsprogrammen im Steuerrechner eines Netzelementes bereitgestellt werden.
- Erfindungsgemäß wird diese technische Aufgabe durch die Merkmale des 3. kennzeichnenden Teils des Anspruchs 1 derart gelöst, daß im Steuerrechner eines Netzelements ein Schnittstellenprogramm bereitgestellt wird, welches bei Eintreffen einer von einem Bedienrechner an ein Anwendungsprogramm in diesem Steuerrechner gerichteten Nachricht, das in dieser Nachricht enthaltene und dem Bedienrechner bekannte Klassenkennzeichen eines Objekts in ein Ersatzkennzeichen umwandelt, welches eine Ersatzklasse des Objekts im Anwendungsprogramm angibt. Das Objekt wird somit bei Abarbeiten einer Nachricht im Anwendungsprogramm als Objekt der Ersatzklasse betrachtet.

Das erfindungsgemäße Verfahren hat den Vorteil, daß die Funktion der Zuordnung von Ersatzkennzeichen zu Klassenkennzeichen nicht in jedem

INTERNATIONALER VORLÄUFIGER PRÜFUNGSBERICHT - BEIBLATT



Anwendungsprogramm, sondern nur einmal an zentraler Stelle im **Schnittstellenprogramm** programmiert werden muß.

- 4. Die erfindungsgemäße Definition des "<u>Verfahrens</u>", wie es in den Merkmalen des kennzeichnenden Teils des **Anspruchs 1** festgelegt ist, wird durch den Stand der Technik weder **offenbart** noch **nahegelegt**:
- 4.1 Dem genannten X.720-Standard sind keine Implementierungshinweise zu entnehmen, die einen Fachmann zu einer Lösung gemäß **Anspruch 1** mit einem <u>Schnittstellenprogramm</u> zur Umsetzung einer <u>Klasse</u> in eine <u>Ersatzklasse</u> leiten würden.
- 4.2 Dem Dokument **D1** sind weder Hinweise auf die <u>Zuordnung von Ersatzklassen</u>, noch auf die Implementierung eines den Anwendungsprogrammen vorgelagerten <u>Schnittstellenprogramms</u> zur Umsetzung von <u>Klassenkennzeichen</u> in <u>Ersatzkennzeichen</u> zu entnehmen.

Der **Anspruch 1** erfüllt daher die Erfordernisse des Artikels 33(2) und (3) PCT hinsichtlich **Neuheit** sowie **erfinderischer Tätigkeit**.

- 5. Die abhängigen **Ansprüche 2-15**, alle direkt oder indirekt von **Anspruch 1** abhängig, erfüllen folglich ebenfalls die Erfordernisse des Artikels 33(2) und (3) PCT hinsichtlich **Neuheit** sowie **erfinderischer Tätigkeit**.
- 6. Der auf ein "Netzelement zum Betreiben eines Telekommunikationsnetzes" gerichtete Anspruch 16 enthält durch die Bezugnahme auf die vorhergehenden Verfahrensansprüche, d.h. durch das Merkmal "..., und daß der Steuerrechner durch das Abarbeiten der Programme das Verfahren nach einem der vorhergehenden Ansprüche durchführt", auch die erfinderischen Merkmale des unabhängigen Anspruchs 1 und erfüllt folglich ebenfalls die Erfordernisse des Artikels 33(2) und (3) PCT hinsichtlich Neuheit sowie erfinderischer Tätigkeit.
- 7. Der auf ein "<u>Telekommunikationsnetz</u>" gerichtete **Anspruch 17** erfüllt aufgrund seiner Abhängigkeit von **Anspruch 16** ebenfalls die Erfordernisse des Artikels 33(2) und (3) PCT hinsichtlich **Neuheit** sowie **erfinderischer Tätigkeit**.

bau solcher offenen Systeme. Zum Führen des Tk-Netzes soll ein separates Führungsnetz verwendet werden. Die Schnittstellen zwischen Bedienrechner und Vermittlungseinrichtung sind in Protokollen Q1, Q2 und Q3 standardisiert.

5

Die Anwendungsobjekte sind als Objekte einer objektorientierten Sprache definiert, z.B. in der Sprache C++ oder CHILL. Werden die Anwendungsprogramme weiterentwickelt, so muß gewährleistet werden, daß das Führungsnetz auch mit den neuen Anwendungsprogrammen fehlerfrei arbeitet. Das bedeutet insbesondere, daß Anwendungsobjekte, die vom Bedienrechner als zu einer ursprünglichen Klasse gehörend angesehen werden, nicht ohne weiteres einer geänderten Ersatzklasse zugeordnet werden können.

15

35

10

Dieses Problem wird im CCITT-Standard X.720 (01/92) - "Information Technology - Open Systems Interconnection - Structure of Management Information: Management Information Model" - im Abschnitt 5.2.1 angesprochen. Im Abschnitt 5.2.3 des Standards X.720 werden zwei Methoden zum Lösen des Problems vor-20 gegeben. Bei der ersten Methode wird auf der Seite des Anwendungsprogramms eine Programmiertechnik verwendet, die Allomorphie berücksichtigt. Allomorphie ist die Fähigkeit eines bestimmten Anwendungsobjektes der Ersatzklasse so geführt zu werden, als wäre es ein Objekt der ursprünglichen Klasse, 25 wenn diese Fähigkeit durch Maßnahmen auf der Seite des Anwendungsprogramms entsteht. Die andere Methode besteht darin, daß auf der Seite des Bedienrechners Maßnahmen getroffen werden, welche auch bei einer Weiterentwicklung des Anwendungsprogramms ein Zusammenarbeiten zwischen Bedienrechner und An-30 wendungsprogramm ermöglichen.

Aus der europäischen Patentanmeldung EP-A-0 817 422 ist ein Verfahren zum Implementieren von geführten Objekten in ein Teilsystem eines geführten Systems in einem Netzwerk bekannt, wobei mindestens ein Führungssystem und ein geführtes System vorhanden sind. Die geführten Objekte werden unabhängig von

anderen Teilsystemen implementiert, ohne den Typ der Objekte in den anderen Teilsystemen zu kennen. Sie können mit anderen Objekten verbunden sein sowie Nachrichten zu ihnen übertragen. Dazu wird ein erstes Objekt zur Kooperation mit einem abstrakten Objekt erzeugt. Das abstrakte Objekt hat eine definierte Schnittstelle, die mit Hilfe des ersten Objekts aufgerufen wird und die einem zweiten Objekt das abstrakte Objekt vererbt, wobei das erzeugte zweite Objekt unbekannt mit dem ersten ist und wobei das zweite Objekt zur Kooperation mit dem ersten Objekt bestimmt ist. Das erste Objekt, das mit dem zweiten Objekt kooperiert, betrachtet das zweite Objekt wie ein Objekt des abstrakten Typs. Jedoch wird bei diesem aus der EP-A- O 817 422 bekannten Verfahren keine Programmiertechnik verwendet, die die Allomorphie gemäß des Abschnittes 5.2.3 des Standards X.720 berücksichtigt.

Es ist Aufgabe der Erfindung zum Betreiben eines Telekommunikationsnetzes ein einfaches Verfahren anzugeben, bei dem Allomorphie berücksichtigt wird.

20

15

5

10

Patentansprüche

- 1. Verfahren zum Betreiben eines Telekommunikationsnetzes (10, 12),
- 5 bei dem ein Netzelement (16) an einem Netzknoten eines Telekommunikationsnetzes (12) von einem Steuerrechner (36) gesteuert wird,

im Steuerrechner (36) neben dem Betriebssystem mindestens ein Schnittstellenprogramm (100) sowie mehrere Anwendungsprogram-

me (102, 104) gespeichert sind, bei deren Ausführen Anwendungsobjekte (a2, a3) bearbeitet werden, die Anwendungsobjekte (a2, a3) je nach Zugehörigkeit zu einer

Klasse (A, A') Daten mit einer vorgegebenen Datenstruktur sowie vorzugsweise auch vorgegebene Verfahren zum Bearbeiten

- der Daten haben,
 zwischen einem Bedienrechner (24) und dem Steuerrechner (36)
 eine Verbindung aufgebaut wird, über die der Steuerrechner
 (36) mittels mindestens einer Wartungsnachricht (WN1 bis WN3)
 gewartet wird, wobei das Schnittstellenprogramm (100) die vom
- Bedienrechner (24) kommenden Wartungsnachrichten (WN1 bis WN3) bearbeitet,
 die Wartungsnachricht (WN1 bis WN3) ein Klassenkennzeichen (moC) enthält, das die Wartungsnachricht (WN1 bis WN3) einer

Klasse (A, A') zuordnet,

- das Klassenkennzeichen (moC) der Wartungsnachricht (WN1 bis WN3) die im Bedienrechner bekannte Klasse (A) eines zu bearbeitenden Anwendungsobjektes (a2, a3) angibt, beim Abarbeiten des Schnittstellenprogramms (100) anhand des Klassenkennzeichens (moC) ein Ersatzkennzeichen ermittelt
- wird, welches eine Ersatzklasse (A') angibt, der das zu bearbeitende Anwendungsobjekt (a2, a3) im Netzelement (16) angehört,
 - beim Abarbeiten des Schnittstellenprogramms (100) das Ersatzkennzeichen (A') in eine geänderte Wartungsnachricht (WN1'
- 35 bis WN2') aufgenommen wird,

und bei dem Bearbeiten der geänderten Wartungsnachricht (WN1'bis WN2') durch ein Anwendungsprogramm (102) das zu bearbeitende Anwendungsobjekt (a2, a3) als Objekt der Ersatzklasse (A') bearbeitet wird.

5

10

- 2. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß beim Ermitteln des Ersatzkennzeichens (A') eine im Speicher (122) des Steuerrechners (16) gespeicherte erste Tabelle (T) verwendet wird, in der dem Klassenkennzeichen (A) ein Ersatzkennzeichen (A') zugeordnet ist.
- 3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß das Anwendungsprogramm (102) nach dem Bearbeiten der geänderten Wartungsnachricht (WN1') eine Bestätigungsnachricht
 (BN1) erzeugt, in der die beim Erzeugen des zu bearbeitenden Anwendungsobjekts (a2, a3) angegebene Klasse (oC) als Klassenkennzeichen (moC) angegeben ist.
- 4. Verfahren nach Anspruch 3, dadurch gekennzeichnet, daß
 20 beim Abarbeiten des Schnittstellenprogramms (100) aus der
 Bestätigungsnachricht (BN1) eine geänderte Bestätigungsnachricht (BN1') erzeugt wird, die nur solche Daten enthält, die
 ein Anwendungsobjekt (a2) der Klasse (A) hat, auf die sich
 die Bestätigungsnachricht (BN1) bezieht,
- 25 und daß die geänderte Bestätigungsnachricht (BN1') an den Bedienrechner (24) gesendet wird.
- 5. Verfahren nach Anspruch 3 oder 4, dadurch gekennzeichnet, daß die beim Erzeugen des zu bearbeitenden Anwendungsobjektes (a2, a3) angegebene Klasse (A, A') als Ursprungsklasse (oC) in den Daten des zu bearbeitenden Anwendungsobjekts (a2, a3) gespeichert ist, und daß beim Abarbeiten des Anwendungsprogramms (102) die Ursprungsklasse (oC) als Klassenkennzeichen (moC) verwendet
- 35 wird.

30

35

- 6. Verfahren nach einem der Ansprüche 3 bis 5, dadurch gekennzeichnet, daß die Bestätigungsnachricht (BN2) ein Kennzeichen (allo) enthält, in welchem mindestens eine Klasse (A)
 bezeichnet ist, die im Bedienrechner (24) als die Klasse (A)
 bekannt ist, zu der das zu bearbeitende Anwendungsobjekt (a2,
 a3) gehört.
- 7. Verfahren nach Anspruch 6, dadurch gekennzeichnet, daß die Bestätigungsnachricht (BN1) neben dem Klassenkennzeichen (moC) ein Ursprungskennzeichen (oC) enthält, in welchem die Ursprungsklasse (A, A') angegeben ist.
- 8. Verfahren nach Anspruch 6 oder 7, dadurch gekennzeichnet, daß mindestens eine im Bedienrechner (24) und/oder in mindestens einem anderen Bedienrechner für das Anwendungsobjekt bekannte Klasse (A) als Allomorphklasse (allo) in den Daten des Anwendungsobjektes (a2, a3) gespeichert ist, und daß beim Abarbeiten des Anwendungsprogramms (102) die Allomorphklasse (allo) als Hilfskennzeichen (allo) verwendet wird.
- 9. Verfahren nach einem der Ansprüche 4 bis 8, dadurch gekennzeichnet, daß die bzw. eine beim Abarbeiten des Schnittstellenprogramms (100) für den Bedienrechner (24) erzeugte

 Bestätigungsnachricht (BN1', BN2') das Klassenkennzeichen
 (moC) und das Ursprungskennzeichen (oC) oder das Klassenkennzeichen (moC), das Ursprungskennzeichen (oC) und ein Kennzeichen (allo), in welchem mindestens eine Klasse (A) bezeichnet
 ist, enthält.
 - 10. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß das Netzelement eine Vermittlungsstelle (16), ein Cross-Connector oder eine Konzentratoreinheit ist.
 - 11. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß das Telekommunikationsnetz (12) ein

20

Festnetz, ein Mobilfunknetz oder ein Netz mit einem Festnetzanteil und einem Mobilfunknetzanteil ist.

- 12. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß das Schnittstellenprogramm (100) weitere Schnittstellenfunktionen zwischen dem Bedienrechner (24) und den Anwendungsprogrammen (102, 104) ausführt, vorzugsweise eine Ereignissteuerung zum Festlegen der Bearbeitungsreihenfolge der Wartungsnachrichten (WN1 bis WN3) und/oder eine Anpassung der vom Bedienrechner (24) kommenden Nachrichten (WN1 bis WN3) an ein Protokoll zur Nachrichtenübertragung innerhalb des Steuerrechners (16), und/oder eine Anpassung der von den Anwendungsprogrammen (102, 104) kommenden Bestätigungsnachrichten (BN1, BN2) an ein vorgegebenes Protokoll zur Nachrichtenübertragung zwischen Bedienrechner (24) und Steuerrechner (16).
 - 13. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß ein erstes Anwendungsprogramm (102) zur Teilnehmerverwaltung, und/oder ein zweites Anwendungsprogramm zum Verwalten von

Verbindungsleitungen zu anderen Vermittlungseinrichtungen, und/oder ein drittes Anwendungsprogramm zur Wartung der Vermittlungseinrichtungen,

- 25 und/oder ein viertes Anwendungsprogramm (104) zur Verkehrsmessung der geschalteten Verbindungen (18) verwendet wird.
- 14. Verfahren nach Anspruch 13, dadurch gekennzeichnet, daß die Anwendungsobjekte (a2, a3) des ersten Anwendungsprogramms (102) für jeweils einen Teilnehmer (Tln2, Tln3) die Teilnehmerdaten enthalten, vorzugsweise die Rufnummer und/oder die nutzbaren Telekommunikationsdienste.
- 15. Verfahren nach einem der vorhergehenden Ansprüche, da-35 durch gekennzeichnet, daß die Wartungsnachrichten (WN1 bis WN3) ein Namenskennzeichen (moI) für den Namen des Anwen-

12-03-2001 | 199901391

dungsobjektes (a2, a3) enthalten, auf welches sich die Wartungsnachricht (WN1 bis WN3) bezieht.

- 16. Netzelement (16) zum Betreiben eines Telekommunikations5 netzes (10, 12) dadurch gekennzeichnet, daß es über einen
 Steuerrechner (36) verfügt, der ein Schnittstellenprogramm
 (100) und mehrere Anwendungsprogramme (102, 104) hat, und daß
 der Steuerrechner (36) durch das Abarbeiten der Programme
 (100, 102, 104) das Verfahren nach einem der vorhergehenden
 10 Ansprüche durchführt.
- 17. Telekommunikationsnetz (10, 12), dadurch gekennzeichnet, daß es eine Vielzahl von Netzelementen enthält und daß es die an die Netzelemente angeschlossenen Teilnehmer verbindet, wobei das Telekommunikationsnetz (10, 12) mindestens ein Netzelement (16) gemäß Anspruch 16 enthält.

of such open systems. A separate control network is to be used to control the telecommunications network. The interfaces between the service computer and switching device are standardized in protocols Q1, Q2 and Q3.

5

The application objects are defined as objects of an object-oriented language, for example in the language C++ or CHILL. If the application programs are developed, it is necessary to ensure that the control network also operates without faults with the new application programs. This means in particular that application objects which are considered by the service computer as belonging to an original class cannot readily be assigned to an amended alternative class.

15

10

This problem is mentioned in the CCITT standard X.720 Technology - Open "Information Interconnection - Structure of Management Information: Management Information Model" - in the section 5.2.1. 20 In the section 5.2.3 of the standard X.720, two methods for solving the problem are proposed. In the first method, on the part of the application program, programming technique is used which takes allomorphy into consideration. Allomorphy is the ability of a specific application object of the alternative class to 25 be controlled as if it were an object of the original class if this ability arises as a result of measures on the part of the application program. The other method comprises taking measures on the part of the service 30 computer which permit cooperation between the service computer and the application program even when the application program is developed.

The object of the invention is to disclose a simple method for operating a telecommunications network in which allomorphy is taken into consideration.

Patent claims

telecommunications for operating a 1. method network (10, 12), in which a network element (16) 5 at a network node of a telecommunications network (12) is controlled by a control computer (36), a plurality of application programs (102, 104) are stored in the control computer (36) in addition to the operating system, during the execution of 10 which application programs (102, 104) application objects (a2, a3) are processed, the application depending objects (a2, a3) having, on membership of a class (A, A'), data with predefined data structure and preferably 15 predefined methods for processing the data, a link via which maintenance is performed on the control computer (36) by means of at least one maintenance message (WN1 to WN3) is set up between a service computer (24) and the control computer (36), the 20 maintenance message (WN1 to WN3) contains a class identifier (moC) which assigns the maintenance message (WN1 to WN3) to a class (A, A'), the class identifier (moC) of the maintenance message (WN1 to WN3) specifies the class (A), known in the 25 service computer, of an application object (a2, a3) to be processed, when the interface program (100) is processed with reference to the class identifier (moC) an alternative identifier determined which indicates an alternative class (A') to which the application object (a2, a3) to 30 be processed belongs in the network element (16), when the interface program (100) is processed the alternative identifier (A') is incorporated into an amended maintenance message (WN1' to WN2'), and when the amended maintenance message 35 in which, the (WN1' to WN2') is processed, application object (a2, a3) to be processed is processed as an object of the alternative class (A') application program (102).

5

- 2. The method as claimed in one of the preceding claims, characterized in that, when the alternative identifier (A') is determined, a first table (T) which is stored in the memory (122) of the control computer (16) is used, in which table (T) an alternative identifier (A') is assigned to the class identifier (A).
- 3. The method as claimed in claim 1 10 characterized in that, after the processing of the amended maintenance message (WN1'), application program (102) generates a confirmation message (BN1) in which the class (oC) specified when the application object (a2, a3) to 15 processed was generated is specified.
- The method as claimed in claim 3, characterized in 4. the interface program when processed, an amended confirmation message (BN1') 20 is generated from the confirmation message (BN1), amended confirmation said message containing only data which has an application object (a2) of the class (A) to which the confirmation message (BN1) relates, and in that 25 amended confirmation message (BN1') the transmitted to the service computer (24).
- 5. The method as claimed in claim 3 characterized in that the class (A, A') specified the application object (a2, a3) 30 processed is generated is stored as an origin class (oC) in the data of the application object (a2, a3) to be processed, and in that, when the application program (102) is processed, the origin class (oC) is used as a class identifier (moC). 35
 - 6. The method as claimed in one of claims 3 to 5, characterized in that the confirmation message (BN1) contains an auxiliary identifier (allo) in

which at least one class (A) is designated which is known in the service computer (24) and/or in at least one other service computer as the

class (A) to which the application object (a2, a3) to be processed belongs, preferably at least the class (A), known in the service computer (24), of the application object (a2, a3) to be processed.

5

10

- 7. The method as claimed in claim 6, characterized in that the confirmation message (BN1) contains, in addition to the class identifier (moC), an origin identifier (oC) in which the origin class (A, A') is specified.
- The method claimed in claim 6 7, 8. as orcharacterized in that at least one class (A) known in the service computer (24) and/or in at least one other service computer for the application 15 object is stored as an allomorph class (allo) in the data of the application object (a2, a3), and in that, when the application program (102) processed, the allomorph class (allo) is used as 20 an auxiliary identifier (allo).
- 9. The method as claimed in one of claims 4 to 8, characterized in that the or a confirmation message (BN1') generated for the service computer (24) when the interface program (100) is being processed contains the auxiliary identifier (allo) and/or the class identifier (moC) and/or the origin identifier (oC).
- 30 10. The method as claimed in one of the preceding claims, characterized in that the network element is a switching office (16), a cross-connector or a concentrator unit.
- 35 11. The method as claimed in one of the preceding claims, characterized in that the telecommunications network (12) is a fixed network, a mobile radio network or a network with a fixed

GR 99 P 1391

network component and a mobile radio network component.

15

The method as claimed in one of the preceding 12. characterized in that the interface further program (100) carries out interface functions between the service computer (24) 5 the application programs (102, 104), preferably an event control for defining processing sequence of the maintenance messages (WN1 to WN3) and/or an adaptation of the messages (WN1 to WN3) coming from the service computer (24) 10 to a protocol for transmitting messages within the control computer (16), and/or an adaptation of the confirmation messages (BN1, BN2) coming from the application programs (102, 104) to a predefined protocol for transmitting messages between the

service computer (24) and control computer (16).

- The method as claimed in one of the preceding 13. claims, characterized in that a first application program (102)is used for subscriber 20 administration, and/or a second application program is used for administering connecting lines to other switching devices, and/or a third application program is used for performing maintenance on the switching devices, 25 and/or a fourth application program (104) is used for measuring the traffic on the switched links (18).
- 30 14. The method as claimed in claim 13, characterized in that the application objects (a2, a3) of the first application program (102) contain the subscriber data, preferably the call number and/or the useable telecommunications services, for one subscriber (Tln2, Tln3) in each case.
 - 15. The method as claimed in one of the preceding claims, characterized in that the maintenance messages (WN1 to WN3) contain a name identifier

GR 99 P 1391

(moI) for the name of the application object (a2, a3) to which the maintenance message (WN1 to WN3) relates. 5

- 16. A network element (16) for operating a telecommunications network (10, 12), characterized in
 that it has a memory for storing instructions
 during whose processing the method according to
 one of the preceding claims is carried out.
- 17. A telecommunications network (10, 12) characterized by a network element as claimed in claim 16.

JC16 Rec'd PCT/PTO SFP 1 0 2001

5/prt> 1

Beschreibung

Verfahren und Netzelement zum Betreiben eines Telekommunikationsnetzes

5

Die Erfindung betrifft ein Verfahren zum Betreiben eines Telekommunikationsnetzes, kurz Tk-Netz, bei dem ein Netzelement an einem Netzknoten des Tk-Netzes von einem Steuerrechner gesteuert wird. Das Netzelement ist beispielsweise eine Vermittlungsstelle zum Vermitteln von Verbindungen, ein soge-10 nannter Cross-Connector oder eine Konzentratoreinheit zum Anschluß mehrerer Teilnehmer an eine Verbindungsleitung. Im Steuerrechner sind neben dem Betriebssystem zum Betreiben des Steuerrechners mehrere Anwendungsprogramme gespeichert, bei 15 deren Ausführen Anwendungsobjekte bearbeitet werden. Zu den Anwendungsobjekten gehören Daten mit einer vorgegebenen Datenstruktur sowie vorzugsweise auch vorgegebene Verfahren zum Bearbeiten der Daten. Die Datenstruktur und die Verfahren sind abhängig von einer auch beim Erzeugen des jeweiligen An-20 wendungsobjektes anzugebenden Klasse. Zwischen einem Bedienrechner und einem Steuerrechner wird eine Verbindung aufgebaut, über die der Steuerrechner mittels Wartungsnachrichten gewartet wird.

25 Derartige Verfahren werden zum Führen des Tk-Netzes eingesetzt, wenn zum Beispiel als Netzelement eine neue Vermittlungseinrichtung in Betrieb genommen wird oder wenn später Teilnehmerdaten in der Vermittlungseinrichtung geändert werden müssen, wie es beim Anschluß neuer Teilnehmer oder beim 30 Umzug eines bisherigen Teilnehmers der Fall ist. Leistungsfähige Verfahren zum Führen des Tk-Netzes entstehen, wenn sogenannte offene Systeme verwendet werden, bei deren Programmierung weltweit geltende Standards beachtet werden. Beispielsweise betreffen Standards der ISO (International Standardisa-35 tion Organization) und der ITU (International Telecommunication Union) mit ihrem Organ ITU-T, früher CCITT (International Telegraph and Telephone Consultativ Committee), den Aufbau solcher offenen Systeme. Zum Führen des Tk-Netzes soll ein separates Führungsnetz verwendet werden. Die Schnittstellen zwischen Bedienrechner und Vermittlungseinrichtung sind in Protokollen Q1, Q2 und Q3 standardisiert.

5

Die Anwendungsobjekte sind als Objekte einer objektorientierten Sprache definiert, z.B. in der Sprache C++ oder CHILL. Werden die Anwendungsprogramme weiterentwickelt, so muß gewährleistet werden, daß das Führungsnetz auch mit den neuen Anwendungsprogrammen fehlerfrei arbeitet. Das bedeutet insbesondere, daß Anwendungsobjekte, die vom Bedienrechner als zu einer ursprünglichen Klasse gehörend angesehen werden, nicht ohne weiteres einer geänderten Ersatzklasse zugeordnet werden können.

15

35

10

Dieses Problem wird im CCITT-Standard X.720 (01/92) - "Information Technology - Open Systems Interconnection - Structure of Management Information: Management Information Model" - im Abschnitt 5.2.1 angesprochen. Im Abschnitt 5.2.3 des Stan-20 dards X.720 werden zwei Methoden zum Lösen des Problems vorgegeben. Bei der ersten Methode wird auf der Seite des Anwendungsprogramms eine Programmiertechnik verwendet, die Allomorphie berücksichtigt. Allomorphie ist die Fähigkeit eines bestimmten Anwendungsobjektes der Ersatzklasse so geführt zu 25 werden, als wäre es ein Objekt der ursprünglichen Klasse, wenn diese Fähigkeit durch Maßnahmen auf der Seite des Anwendungsprogramms entsteht. Die andere Methode besteht darin, daß auf der Seite des Bedienrechners Maßnahmen getroffen werden, welche auch bei einer Weiterentwicklung des Anwendungs-30 programms ein Zusammenarbeiten zwischen Bedienrechner und Anwendungsprogramm ermöglichen.

Es ist Aufgabe der Erfindung zum Betreiben eines Telekommunikationsnetzes ein einfaches Verfahren anzugeben, bei dem Allomorphie berücksichtigt wird.

30

35

Diese Aufgabe wird durch ein Verfahren mit den Merkmalen des Patentanspruchs 1 gelöst. Weiterbildungen sind in den Unteransprüchen angegeben.

5 Beim erfindungsgemäßen Verfahren wird aus den Wartungsnachrichten beim Abarbeiten eines für mehrere Anwendungsprogramme genutzten Schnittstellenprogramms jeweils ein Klassenkennzeichen ermittelt, in welchem die Klasse angegeben ist, auf die sich die Wartungsnachricht bezieht. Das Klassenkennzeichen in 10 der Wartungsnachricht gibt die im Bedienrechner bekannte Klasse eines zu bearbeitenden Anwendungsobjektes an. Durch die Weiterentwicklung kommt es vor, daß die im Bedienrechner bekannte Klasse von der tatsächlichen Klasse des Anwendungsobjekts abweicht. Beim Abarbeiten des Schnittstellenprogramms 15 wird an Hand des Klassenkennzeichens ein Ersatzkennzeichen ermittelt, welches eine Ersatzklasse angibt, der das Anwendungsobjekt im Netzelement zugeordnet ist. Das Ersatzkennzeichen wird in eine geänderte Wartungsnachricht aufgenommen. Beim Bearbeiten der geänderten Wartungsnachricht durch ein 20 Anwendungsprogramm wird das Anwendungsobjekt dann als zur Ersatzklasse gehörend bearbeitet. Dies ist möglich, weil das Anwendungsobjekt allomorph zu der im Bedienrechner bekannten Klasse ist, die in der ungeänderten Wartungsnachricht vom Bedienrechner für das Anwendungsobjekt vorausgesetzt worden 25 ist.

Das Schnittstellenprogramm übernimmt zentral für mehrere Anwendungsprogramme die Zuordnung der Ersatzkennzeichen zu den
Klassenkennzeichen. Beim erfindungsgemäßen Verfahren muß dieser Schritt nicht in jedem Anwendungsprogramm sondern nur
einmal im Schnittstellenprogramm programmiert werden. Bei
mehreren hundert Anwendungsprogrammen je Steuerrechner verringert sich dadurch der Programmier-, Wartungs- und Dokumentationsaufwand erheblich. Die Anwendungsprogamme werden von
zusätzlichen Schritten freigehalten, die beim Berücksichtigen
von Allomorphie notwendig sind, weil diese Schritte zentral
im vorgelagerten Schnittstellenprogramm durchgeführt werden.

35

Ein Teil der zusätzlichen Schritte wird auch in nachgelagerten Datenbanken durchgeführt, welche von den Anwendungsprogrammen genutzt werden.

5 Das Ausführen der Zuordnung von Ersatzkennzeichen und Hilfskennzeichen in einem zentralen Schnittstellenprogramm ist möglich, weil Allomorphie beim erfindungsgemäßen Verfahren auf Klassenebene definiert ist. Eine derartige Definition ist im Standard X.720 nicht erwähnt aber dennoch standardgerecht. 10 Allomorphie auf Klassenebene bedeutet, daß alle Objekte der Ersatzklasse so geführt werden können, als wären sie Objekte der ursprünglichen Klasse. Durch eine auf alle Objekte der Ersatzklasse bezogene Definition von Allomorphie entstehen dann keine Nachteile, wenn vorgegebene Programmierregeln be-15 achtet werden. Beispiele für solche Programmierregeln werden unten im Zusammenhang mit den Ausführungsbeispielen erläutert.

Das erfindungsgemäße Verfahren ermöglicht es, die Vorgaben
20 des Standards X.720 auf eine einfache Art und Weise einzuhalten. Die Anwendungsprogramme im Steuerrechner können mit einem geringen Mehraufwand weiterentwickelt werden, wobei immer
sichergestellt bleibt, daß auch bei unverändert gebliebenen
Programmen im Bedienrechner keine Fehler beim Betreiben des
25 Führungsnetzes auftreten.

In einer Weiterbildung wird im Schnittstellenprogramm eine Tabelle verwendet, mit der dem Klassenkennzeichen die Ersatzkennzeichen zugeordnet werden. Die Tabelle ist im Speicher des Steuerrechners gespeichert. Ein Eintrag der Tabelle wird gelesen, indem eine dem Klassenkennzeichen zugeordnete Speicherzelle gelesen wird, die das zum Klassenkennzeichen gehörende Ersatzkennzeichen enthält. Das Ermitteln des Ersatzkennzeichens benötigt so nur einen einzigen Lesezugriff auf den Speicher. Ändern sich durch Weiterentwicklungen der Anwendungsprogramme die Ersatzkennzeichen, so müssen nur die Speicherinhalte neu programmiert werden. Das bedeutet, daß

der Inhalt der Tabelle leicht ausgetauscht oder erweitert werden kann.

Wird in einer anderen Weiterbildung durch das Anwendungsprogramm nach dem Bearbeiten der geänderten Wartungsnachricht eine Bestätigungsnachricht erzeugt, in der die beim Erzeugen des zu bearbeitenden Anwendungsobjektes angegebene Klasse als Klassenzeichen angegeben ist, so kann die Bestätigungsnachricht vom Schnittstellenprogramm nachfolgend auf einfache Art weiter bearbeitet werden. Beispielsweise kann beim Abarbeiten des Schnittstellenprogramms an Hand des Klassenkennzeichens festgestellt werden, welche Daten aus der Bestätigungsnachricht zu entfernen sind. Dazu wird die im Schnittstellenprogramm verwendete Tabelle so erweitert, daß zu jedem Klassenkennzeichen auch Eintragungen zu den erlaubten Daten gehören. Das Schnittstellenprogramm erzeugt dann aus der Bestätigungsnachricht eine neue Bestätigungsnachricht, die nur solche Daten eines Anwendungsobjekts der Klasse enthält, auf die sich die Bestätigungsnachricht bezieht.

20

25

5

10

15

Die beim Erzeugen des zu bearbeitenden Anwendungsobjektes angegebene Klasse wird in einer Weiterbildung als Ursprungsklasse in den Daten des zu bearbeitenden Anwendungsobjekts gespeichert. Beim Abarbeiten des Anwendungsprogramms wird die Ursprungsklasse dann als Klassenkennzeichen verwendet. Durch diese Vorgehensweise ist die Ursprungsklasse auf einfache Art und Weise verfügbar.

dung auch ein Hilfskennzeichen, in welchem mindestens eine Klasse bezeichnet ist, die im Bedienrechner und/oder in mindestens einem anderen Bedienrechner als die Klasse bekannt ist, zu der das zu bearbeitende Anwendungsobjekt gehört, so kann später das Programm im Bedienrechner an Hand des Hilfskennzeichens ermitteln, wie die empfangene Bestätigungsnachricht zu bearbeiten ist. Dies ist insbesondere dann von Bedeutung, wenn die im Klassenkennzeichen der vom Bedienrechner

empfangenen Bestätigungsnachricht angegebene Klasse im Bedienrechner noch nicht bekannt ist. Der Bedienrechner bestimmt die Klasse, auf die sich die Bestätigungsnachricht bezieht dann an Hand der im Hilfskennzeichen angegebenen Klasse bzw. Klassen. Das Hilfskennzeichen enthält, mit anderen Worten ausgedrückt, die Klassen, zu denen das Anwendungsobjekt allomorph ist. Enthält die Bestätigungsnachricht neben dem Klassenkennzeichen auch ein Ursprungskennzeichen, in dem die Ursprungsklasse angegeben ist, so lassen sich die Vorgaben des Protokolls für den Nachrichtenaustausch im Steuerrechner und auch für das Protokoll für den Nachrichtenaustausch zwischen Bedienrechner und Steuerrechner erfüllen.

In einer anderen Weiterbildung wird mindestens eine im Bedienrechner und/oder in mindestens einem anderen Bedienrechner für das Anwendungsobjekt bekannte Klasse als Allomorphklasse in den Daten des Anwendungsobjekts gespeichert. Beim
Abarbeiten des Anwendungsprogramms wird dann die Allomorphklasse als Hilfskennzeichen verwendet. Durch diese Maßnahme
20 entsteht eine übersichtliche Datenstruktur, bei der die Anwendungsobjekte ihre Allomorphklassen selbst verwalten. Im
Schnittstellenprogramm und im Anwendungsprogramm müssen keine
zusätzlichen Maßnahmen hinsichtlich der Allomorphklasse getroffen werden.

25

30

10

In einer anderen Weiterbildung ist das Schnittstellenprogramm auch für andere Schnittstellenfunktionen zuständig. Beispielsweise für die Ereignissteuerung zum Festlegen der Bearbeitungsreihenfolge der Wartungsnachrichten oder für Protokollanpassungen dieser Nachrichten, englisch als "basic encoding" bezeichnet. Durch diese Maßnahme gibt es auf dem Steuerrechner nur ein einziges Schnittstellenprogramm, das einheitlich programmiert und gewartet wird.

Im folgenden werden Ausführungsbeispiele der Erfindung an Hand der Zeichnungen erläutert. Darin zeigen:

- Figur 1 einen Teil eines Führungsnetzes zum Führen eines Telekommunikationsnetzes,
- Figur 2 die Weiterentwicklung einer ursprünglichen Klasse A zu einer erweiterten Klasse A', deren Objekte beim Betreiben des Führungsnetzes wie Objekte der alten Klasse A geführt werden können,
- Figur 3 das Bearbeiten von Nachrichten im Steuerrechner einer Vermittlungseinheit nach der Weiterentwicklung,
 wobei ein Objekt geführt wird, das vor der Weiterentwicklung erzeugt worden ist,
- Figur 4 das Bearbeiten von Nachrichten im Steuerrechner

 nach der Weiterentwicklung, wobei ein Objekt geführt wird, das nach der Weiterentwicklung erzeugt
 worden ist,
- Figur 5 die Namensbindung der Klassen A und A' sowie ein

 Zugriff auf Objekte der beiden Klassen mittels ein
 ner Filterfunktion.

Figur 1 zeigt einen Teil eines Führungsnetzes 10 zum Führen eines Telekommunikationsnetzes 12, kurz Tk-Netz 12 genannt.

Das Tk-Netz 12 enthält eine Vielzahl von Vermittlungsstellen, von denen in Figur 1 die Vermittlungsstellen 14 und 16 dargestellt sind. Zum Tk-Netz 12 gehören weiterhin Verbindungsleitungen zwischen den Vermittlungsstellen, von denen in Figur 1 eine Verbindungsleitung 18 zwischen der Vermittlungsstelle 14 und der Vermittlungsstelle 16 dargestellt ist. Das Tk-Netz 12 verbindet die Teilnehmer des Tk-Netzes 12, beispielsweise einen an die Vermittlungsstelle 14 angeschlossenen Teilnehmer Tln1 und einen an die Vermittlungsstelle 16 angeschlossenen Teilnehmer Teilnehmer Tln2.

35

Das Führungsnetz 10 enthält eigene Übertragungsstrecken 20 und 22, die zu einem Bedienrechner 24 führen. Die Übertra-

gungsstrecke 20 überträgt Wartungsnachrichten vom Bedienrechner 24 zur Vermittlungsstelle 14, um beispielsweise in der Vermittlungsstelle 14 Teilnehmerdaten des Teilnehmers Tln1 zu ändern. Die Vermittlungsstelle 14 sendet ihrerseits Bestätigungsnachrichten an den Bedienrechner 24, um die ordnungsgemäße Bearbeitung der empfangenen Wartungsnachricht zu signalisieren. Die Übertragungsstrecke 22 dient zur bidirektionalen Datenübertragung zwischen Bedienrechner 24 und Vermittlungsstelle 16.

10

5

Die Wartungsnachrichten werden in der Vermittlungsstelle 14
von einem Steuerrechner 34 und in der Vermittlungsstelle 16
von einem Steuerrechner 36 bearbeitet. Die Datenstrukturen,
auf die sich die Wartungsnachrichten beziehen, gehören im Bedienrechner 24 und in der Vermittlungsstelle 14 derselben
Klasse A an. Die Vermittlungsstelle 16 enthält dagegen Datenstrukturen einer Klasse A', die gegenüber der im Bedienrechner 24 vorausgesetzten Klasse A weiterentwickelt worden ist.
Der fehlerfreie Betrieb des Führungsnetzes 10 wird bezüglich
der Vermittlungsstelle 16 dadurch gewährleistet, daß beim
Weiterentwickeln der Klasse A zur Klasse A' Allomorphie berücksichtigt worden ist. Was Allomorphie in diesem Zusammenhang bedeutet, wird unten an Hand der Figur 2 erläutert.

Beim in Figur 1 dargestellten Ausführungsbeispiel wird in der Vermittlungsstelle 14 die Klasse A verwendet, die beispiels-weise die Datenstruktur für Teilnehmerdaten festlegt, z.B. die Rufnummer und die nutzbaren Dienste des Tk-Netzes 12. Teilnehmerdaten des Teilnehmers Tln1 sind in einem Objekt al gemäß der durch die Klasse A vorgegebenen Datenstruktur in einem Speicher 38 des Steuerrechners 34 gespeichert. Die Klasse A ist auch im Bedienrechner 24 bekannt, angedeutet durch den Buchstaben A in einem Speicher 40 des Bedienrech-

35

ners 24.

In der Vermittlungsstelle 16 wurde die Klasse A zur Klasse A' weiterentwickelt. Ein Objekt a2 enthält beispielsweise die

Teilnehmerdaten des Teilnehmers Tln2. Das Objekt a2 wurde erstmalig im Speicher 42 gespeichert, bevor die Klasse A zur Klasse A' weiterentwickelt worden ist. Bei der Weiterentwicklung wurde das ursprüngliche Objekt a2 aber umgewandelt, und zwar in ein erweitertes Objekt a2 der Klasse A', indem ein Datenfeld ergänzt worden ist. Ein Objekt a3 im Speicher 42 gehört zur Klasse A' und enthält die Teilnehmerdaten eines Teilnehmers Tln3, dessen Anschluß erst nach der Weiterentwicklung in der Vermittlungsstelle 16 eingerichtet worden 10 ist. Obwohl die Programme im Bedienrechner 24 nur Objekte der Klasse A unterstützen, können die zur Klasse A' gehörenden Objekte a2 und a3 vom Bedienrechner 24 aus wie Objekte der Klasse A abgefragt, geändert oder neu eingerichtet werden. Die Erweiterungen der Klasse A' im Vergleich zur Klasse A 15 können vom Bedienrechner 24 allerdings erst dann bearbeitet werden, wenn die Programme im Bedienrechner 24 zu einem späteren Zeitpunkt so geändert werden, daß auch die Klasse A' im Bedienrechner 24 bekannt ist.

Figur 2 zeigt die Klassen A und A' sowie das ursprüngliche 20 Objekt a2 und das Objekt a3. An Hand der Figur 2 wird im folgenden erläutert, was die Bezeichnung "allomorph zu" bedeutet. Die Klasse A' unterscheidet sich von der Klasse A nur durch ein zusätzliches Datenfeld 50. Die Datenstruktur der 25 Klasse A wurde also um das Datenfeld 50 erweitert, um eine weitere Eigenschaft der Teilnehmer Tln2, Tln3 beim Betrieb der Vermittlungsstelle 16 berücksichtigen zu können, z.B. ob der Teilnehmer Tln2, Tln3 über einen Lichtwellenleiter oder. über einen Kupferleiter an die Vermittlungsstelle 16 ange-30 schlossen ist. Die Klasse A' wird deshalb im folgenden auch als erweiterte Klasse A' bezeichnet. Ein Datenfeld 50' im Objekt a3 enthält als Datum einen Wert, der angibt, daß der Teilnehmer Tln3 mittels eines Lichtwellenleiters an die Vermittlungsstelle 16 angeschlossen ist. Das Objekt a3 wird all-35 gemein als erweitertes Objekt a3 bezeichnet.

Die Klasse, die beim Erzeugen eines Objektes angegeben wird, wird als ursprüngliche Klasse dieses Objekts bezeichnet. Das Objekt a2 hatte als ursprüngliche Klasse die Klasse A, angedeutet durch einen Pfeil 52. Das erweiterte Objekt a3 hat dagegen als ursprüngliche Klasse die erweiterte Klasse A', angedeutet durch einen Pfeil 54.

Eine erste Möglichkeit zum Festlegen der Datenstruktur der Klasse A besteht darin, die Klasse A' aus der Klasse A mit-10 tels sogenannter Vererbung zu erzeugen, die in objektorientierten Programmiersprachen definiert ist. Solche Programmiersprachen sind z.B. die Sprachen C++ und CHILL. Bei einer Vererbung wird durch den Programmierer angegeben, daß die erweiterte Klasse A' von der Klasse A sämtliche Datenstrukturen 15 und sogenannten Methoden zum Bearbeiten der Datenstrukturen übernehmen soll. Weiterhin wird angegeben, daß die Klasse A' zusätzlich das Datenfeld 50 enthält. Eine andere Möglichkeit zum Festlegen der Klasse A' besteht darin, diese Klasse neu zu definieren. In diesem Fall wird die Klasse A' so definiert, wie es bereits bei der Klasse A der Fall war. Zusätz-20 lich wird jedoch noch das Datenfeld 50 definiert. Die Beziehung der übereinstimmenden Teile der Klasse A und der erweiterten Klasse A' ist in Figur 2 mittels Strichlinien 56 dargestellt.

25

30

35

5

In einem gedachten Objekt a3* sind aus dem Objekt a3 alle Datenfelder und alle Methoden zum Bearbeiten der Datenfelder enthalten, die auch beim Einrichten des Teilnehmers Tln3 vor der Weiterentwicklung erzeugt worden wären, als es zwar die Klasse A, aber noch nicht die Klasse A' gab. Im Objekt a3* fehlt deshalb ein dem Datenfeld 50 bzw. 50' entsprechendes Datenfeld. Dieser Sachverhalt wird durch Strichlinien 58 verdeutlicht. Das Objekt a3* ist eine Anschauungshilfe zum Abgrenzen der Begriffe "kompatibel zu" und "allomorph zu". Ein Pfeil 60 verdeutlicht, daß das Objekt a3* kompatibel zur Klasse A ist, weil es genau die Datenstrukturen hat, die in der Klasse A vorgegeben sind. Das erweiterte Objekt a3 ist

25

30

dagegen allomorph zur Klasse A, vgl. Pfeil 62. Das Objekt a3 hat die allomorphe Klasse A.

Allomorphie ist die Fähigkeit der Objekte der Klasse A' so geführt zu werden, als wären sie Objekte ihrer allomorphen Klasse A, wenn diese Fähigkeit durch Maßnahmen auf der Seite des Anwendungsprogramms entsteht. Bei einer stufenweisen Erweiterung kann es mehr als eine allomorphe Klasse geben, z.B. die allomorphe Klasse der letzten Erweiterung und die allomorphe Klasse der vorletzten Erweiterung.

Ein erweitertes Objekt hat nur dann eine allomorphe Klasse, wenn das erweiterte Objekt ohne die Erweiterungen zur allomorphen Klasse kompatibel gemäß Standard X.720 Abschnitt

5.2.2 ist. Insbesondere hat das erweiterte Objekt demnach alle Attribute, Attributgruppen, Führungsfunktionen und Bestätigungsverfahren, die auch in der allomorphen Klasse festgelegt sind. Durch das Berücksichtigen der Allomorphie beim Erweitern der Klasse A wird erreicht, daß das Führungsnetz 10 auch nach der Erweiterung fehlerfrei arbeitet.

Figur 3 zeigt das Bearbeiten von Nachrichten im Steuerrechner 16 nach der Weiterentwicklung der Klasse A zur Klasse A', wobei das Objekt a2 geführt wird, das vor der Weiterentwicklung als zur Klasse A gehörend angelegt worden ist. Beim Weiterentwickeln der Klasse A zur Klasse A' wurde Allomorphie berücksichtigt, so daß Objekte der Klasse A' zur Klasse A allomorph sind. Außerdem wurden beim Weiterentwickeln alle Objekte der Klasse A in Objekte der Klasse A' umgewandelt, indem Datenfelder und Methoden ergänzt worden sind.

Der Steuerrechner 16 enthält ein Schnittstellenprogramm 100, das vom Bedienrechner 24 kommende Wartungsnachrichten, beispielsweise eine Wartungsnachricht WN1, bearbeitet und das Bestätigungsnachrichten, beispielsweise eine Bestätigungsnachricht BN1', an den Bedienrechner 24 sendet. Das Schnittstellenprogramm 100 ist die Schnittstelle zwischen dem Bedienrechner 24 und mehreren Anwendungsprogrammen im Steuerrechner 16, von denen in Figur 3 zwei Anwendungsprogramme 102
und 104 gezeigt sind. Das Anwendungsprogramm 102 dient zum
Verwalten der zu den an die Vermittlungsstelle 16 angeschlossenen Teilnehmern Tln2, Tln3 gehörenden Daten. Das Anwendungsprogramm 104 wird zur Verkehrsmessung verwendet.

Die vom Bedienrechner kommende Wartungsnachricht WN1 wird beim Abarbeiten des Schnittstellenprogramms 100 an das Anwendungsprogramm 102 als geänderte Wartungsnachricht WN1' weitergeleitet. Für das Anwendungsprogramm 104 bestimmte Wartungsnachrichten werden dagegen beim Abarbeiten des Schnittstellenprogramms 100 an das Anwendungsprogramm 104 weitergeleitet, vgl. Pfeil 106.

15

20

25

30

35

10

5

Nach dem Bearbeiten der Wartungsnachricht WN1' im Anwendungsprogramm 102 erzeugt das Anwendungsprogramm 102 eine Bestätigungsnachricht BN1 für die Schnittstelle 100. Hat das Anwendungsprogramm 104 eine Wartungsnachricht bearbeitet, so sendet es ebenfalls eine Bestätigungsnachricht an das Schnittstellenprogramm 100, vgl. Pfeil 108.

Beim Bearbeiten der Wartungsnachricht WN1' arbeitet das Anwendungsprogramm 102 mit einem ebenfalls im Steuerrechner 36 vorhandenen Datenbankprogramm 110 zusammen, mit dem Teilnehmerdaten im Speicher 42 gespeichert, verändert, gelöscht oder gelesen werden. Das Anwendungsprogramm 102 sendet Anforderungen in Form von Nachrichten an das Datenkbankprogramm 110, beispielsweise die Nachricht N1. Nach dem Ausführen der Anforderung in der Nachricht N1 sendet das Datenbankprogramm 110 eine Ergebnisnachricht EN1 an das Anwendungsprogramm 102 zurück. Das Anwendungsprogramm 104 arbeitet mit einem eigenen Datenbankprogramm 112 zusammen, indem es Anforderungen an das Datenbankprogramm 112 sendet, vgl. Pfeil 114, und indem es vom Datenbankprogramm 112 Ergebnisnachrichten empfängt und

weiterbearbeitet, vgl. Pfeil 116.

Das Datenbankprogramm 110 benutzt für die Objekte der ursprünglichen Klasse A und die Objekte der weiterentwickelten Klasse A' dasselbe Zugriffsverfahren. Dies ist möglich, weil Kombinationsklassen verwendet werden, in denen die Datenstrukturen und Methoden der ursprünglichen Klasse und der erweiterten Klasse zusammengeführt sind. Eine Kombinationsklasse KA ist die Kombination der Klasse A und der Klasse A'. Sämtliche Objekte im Speicher 42, welche die allomorphe Klasse A als ursprüngliche Klasse haben, werden mit der Weiterentwicklung um die zusätzlichen Datenfelder der erweiterten Klasse A erweitert. Die zusätzlichen Datenfelder werden mit vorgegebenen Werten belegt.

Die Wartungsnachricht WN1 enthält ein Klassenkennzeichen moC,
das die Klasse A als die Klasse angibt, auf welche sich die
Wartungsnachricht WN1 bezieht. Ein Objektkennzeichen moI gibt
das Objekt a2 an, auf das sich die Wartungsnachricht WN1 bezieht. Die Wartungsnachricht WN1 wird vom Bedienrechner 24
gesendet, um die Teilnehmerdaten des Teilnehmers Tln2 zu erfahren. Im Bedienrechner 24 ist lediglich bekannt, daß diese
Teilnehmerdaten in dem Objekt a2 enthalten sind, das im Steuerrechner 36 gespeichert ist. Die Wartungsnachricht WN1 enthält weitere nicht dargestellte Datenfelder. In einem dieser
Datenfelder ist beispielsweise die auszuführende Leseoperation festgelegt.

Beim Bearbeiten der Wartungsnachricht WN1 im Schnittstellenprogramm 100 wird die im Klassenkennzeichen moC angegebene
Klasse A ermittelt. An Hand der so ermittelten Klasse A wird
beim Abarbeiten des Schnittstellenprogramms 100 mittels einer
Tabelle T die Klasse A' als Ersatzklasse ermittelt und in das
Klassenkennzeichen moC der Wartungsnachricht WN1 eingetragen.
Die Tabelle T ist in einem Speicher 122 des Steuerrechners 36
gespeichert.

35

30

10

Die Wartungsnachricht WN1' betrifft mit dem Klassenkennzeichen moC=A' die erweiterte Klasse A'. Der Wert des Objekt-

10

kennzeichens moI=a2 bleibt in der Wartungsnachricht WN1' unverändert. Ein Kennzeichen allo in der Wartungsnachricht WN1' gibt alle Klassen an, die zur Klasse A' allomorph sind, d.h. im Ausführungsbeispiel die Klasse A. Das Schnittstellenprogramm 100 entnimmt diese Klassen ebenfalls der Tabelle T. Die anderen Daten der Wartungsnachricht WN1 werden in die Wartungsnachricht WN1' übernommen. Beim Erzeugen der zur Wartungsnachricht WN1' gehörenden Wartungsnachricht WN1' führt das Schnittstellenprogramm 100 auch eine Protokollanpassung von einem Übertragungsprotokoll auf der Übertragungsstrecke 22 in ein Nachrichtenprotokoll durch, das innerhalb des Steuerrechners 36 verwendet wird.

Die beim Bearbeiten der Wartungsnachricht WN1' durch das Anwendungsprogramm 102 erzeugte Nachricht N1 enthält einen Befehl B1, der angibt, daß Daten gelesen werden sollen. Als Parameter des Befehls B1 enthält die Nachricht N1 die Klasse
A', zu der die zu lesenden Daten gehören, sowie das Objekt
a2, dessen Daten gelesen werden sollen. Das Anwendungsprogramm 102 bearbeitet ausschließlich Nachrichten, die sich auf
Objekte der Klasse A' beziehen. Bezüglich der Klasse A sind
im Anwendungsprogramm 102 keine weiteren Maßnahmen getroffen.

Das Datenbankprogramm 110 greift beim Bearbeiten der Nach-25 richt N1 auf den Speicher 42 zu, um die Daten des Teilnehmers Tln2 zu lesen, die im Objekt a2 gespeichert sind. Das Objekt a2 enthält außerdem ein Ursprungskennzeichen oC, in dem die Klasse A angegeben ist, die beim Erzeugen des Objekts a2 angegeben worden ist. Das Datenbankprogramm 110 liest die mit 30 dem Befehl B1 angeforderten Daten im Objekt a2 und trägt diese Daten in die Ergebnisnachricht EN1 ein. In der Ergebnisnachricht EN1 ist außerdem mittels eines Antwortkennzeichens AB1 vermerkt, daß die Ergebnisnachricht EN1 das Ergebnis enthält, das beim Bearbeiten des Befehls B1 erzeugt worden ist. 35 Weiterhin ist in der Ergebnisnachricht EN1 die Klasse A' als die Klasse angegeben, welche von der Ergebnisnachricht EN1 betroffen ist. Das Ursprungskennzeichen oC=A wird ebenfalls

vom Datenbankprogramm 110 in der Ergebnisnachricht EN1 an das Anwendungsprogramm 102 übermittelt.

Beim Abarbeiten des Anwendungsprogramms 102 wird vorausgesetzt, daß sich alle zu bearbeiteten Nachrichten auf die
Klasse A' und nicht auf die Klasse A beziehen. Durch das Anwendungsprogramm 102 wird der Wert des Ursprungskennzeichens
oC=A als Wert des Klassenkennzeichens moC in der Bestätigungsnachricht BN1 übernommen. Für diese Wertzuweisung ist es
nicht erforderlich, daß das Anwendungsprogramm 102 Objekte
der Klasse A bearbeiten kann. Außerdem enthält die Bestätigungsnachricht BN1 das Ursprungskennzeichen oC=A und die abgefragten Teilnehmerdaten des Teilnehmers Tln2. Das Schnittstellenprogramm 100 braucht sich aufgrund dieser Vorgehensweise das Klassenkennzeichen der Wartungsnachricht WN1 nicht
zu merken.

Die Bestätigungsnachricht BN1 wird vom Schnittstellenprogramm 100 bearbeitet und als Bestätigungsnachricht BN1' gemäß dem Übertragungsprotokoll auf der Übertragungsstrecke 22 an den Bedienrechner 24 gesendet. An Hand der Tabelle T ermittelt das Schnittstellenprogramm 100, welche Datenfelder in der Nachricht BN1 nicht in Objekten der Klasse A enthalten sind. Diese Datenfelder werden nicht in die Bestätigungsnachricht 25 BN1' übernommen.

Der Bedienrechner 24 empfängt die Bestätigungsnachricht BN1' und kann sie wie eine Nachricht bearbeiten, die sich auf ein Objekt der Klasse A bezieht. Objekte der Klasse A' im Steuerrechner 36 werden vom Bedienrechner 24 aus so geführt, als wären sie Objekte der Klasse A. Die Weiterentwicklung im Steuerrechner 36 schränkt die Betriebsmerkmale des Bedienrechners 24 somit nicht ein.

Figur 4 zeigt das Bearbeiten von Nachrichten im Steuerrechner 16 nach der Weiterentwicklung der Klasse A zur Klasse A', wobei das Objekt a3 geführt wird, das erst nach der Weiterent-

wicklung erzeugt wird. Durch die Eingaben einer Bedienperson in den Bedienrechner 24 wird im Bedienrechner 24 eine Wartungsnachricht WN2 erzeugt, welche die Aufgabe hat, im Steuerechner 16 das Objekt a3 für die Teilnehmerdaten des Teilnehmers Tln3 zu erzeugen. Die Wartungsnachricht WN2 enthält deshalb in einem Befehlsfeld eine Codierung für den Befehl "Erzeuge". Das Klassenkennzeichen moC in der Wartungsnachricht WN2 kennzeichnet die Klasse A als die Klasse, auf die sich die Wartungsnachricht WN2 bezieht. Das Objektkennzeichen moI der Wartungsnachricht WN2 kennzeichnet das Objekt a3, das durch die Wartungsnachricht WN2 erzeugt werden soll. Die Wartungsnachricht WN2 enthält außerdem noch weitere nicht dargestellte Daten.

Das Schnittstellenprogramm 100 bearbeitet die Wartungsnach-15 richt WN2 ebenso wie oben für die Wartungsnachricht WN1 erläutert. Beim Bearbeiten der Wartungsnachricht WN2 wird wiederum die im Speicher 122 gespeicherte Tabelle T verwendet, um für die im Klassenkennzeichen moC der Wartungsnachricht 20 WN2 angegebene Klasse A die Ersatzklasse A' zu ermitteln und als Klassenkennzeichen moC der Wartungsnachricht WN2' zu verwenden. Das Objektkennzeichen moI=a3 hat in der Wartungsnachricht WN2' den gleichen Wert, wie in der Wartungsnachricht WN2. Auch die verbleibenden Daten werden aus der Wartungs-25 nachricht WN2 beim Ausführen des Schnittstellenprogramms 100 in die Wartungsnachricht WN2' übernommen. Zusätzlich wird im Kennzeichen allo der Wartungsnachricht WN2' vermerkt, daß die Klasse A die Klasse ist, zu der die Klasse A' allomorph ist. Die so erzeugte Wartungsnachricht WN2' wird vom Schnittstel-30 lenprogramm 100 gemäß dem Protokoll im Steuerrechner 36 an das Anwendungsprogramm 102 gesendet.

Das Anwendungsprogramm 102 sendet beim Bearbeiten der Wartungsnachricht WN2' eine Nachricht N2 an das Datenbankprogramm 110, um das Objekt a3 im Speicher 42 anlegen zu lassen. Die Nachricht N2 enthält verschlüsselt den Befehl "Erzeuge",

15

20

25

die erweiterte Klasse A' sowie den Namen des anzulegenden Objekts a3.

Beim Bearbeiten der Nachricht N2 mittels des Datenbankprogramms 110 wird im Speicher 42 das Objekt a3 erzeugt, indem diesem Objekt bestimmte Speicherzellen zugewiesen werden, die mit Anfangswerten belegt werden. Im Ursprungskennzeichen oC des Objekts a3 wird die Klasse A' vermerkt, weil diese Klasse beim Erzeugen des Objektes a3' angegeben wurde. Ein Kennzeichen allo im Objekt a3 weist auf die Klasse A hin, weil das Objekt a3 zu der Klasse A allomorph ist.

Das Datenbankprogramm 110 erzeugt eine Ergebnisnachricht EN2, um das Erzeugen des Objekts a3 zu bestätigen. Die Ergebnisnachricht EN2 enthält ein Antwortkennzeichen AB2, das angibt, daß die Ergebnisnachricht EN2 beim Bearbeiten eines Befehls "Erzeuge" entstanden ist. Weiterhin enthält die Ergebnisnachricht EN2 ein Kennzeichen, das auf die Klasse A' hinweist, das Ursprungskennzeichen oC=A', das Kennzeichen allo=A sowie weitere nicht dargestellte Daten des Objekts a3.

Die Ergebnisnachricht EN2 wird vom Anwendungsprogramm 102 bearbeitet, wobei eine Bestätigungsnachricht BN2 erzeugt wird. In der Bestätigungsnachricht BN2 wird als Wert des Klassenzeichens moC der Wert des Ursprungskennzeichens oC=A' verwendet. Die anderen Daten der Ergebnisnachricht EN2 werden in die Bestätigungsnachricht BN2 übernommen.

Beim Abarbeiten des Schnittstellenprogramms 100 wird nach dem Empfang der Bestätigungsnachricht BN2 eine Bestätigungsnachricht BN2' gemäß dem auf der Übertragungsstrecke 22 verwendeten Übertragungsprotokoll erzeugt. Die Bestätigungsnachricht BN2' enthält alle Daten der Bestätigungsnachricht BN2, weil das Schnittstellenprogramm 100 an Hand der Tabelle T feststellt, daß keine Datenfelder entfernt werden müssen, wenn die Bestätigungsnachricht BN2 als Klassenkennzeichen die Klasse A' hat.

Beim Bearbeiten der Bestätigungsnachricht BN2' im Bedienrechner 24 wird an Hand des Kennzeichens allo festgestellt, daß die Bestätigungsnachricht BN2' Daten eines Objekts der Klasse 5 A betrifft bzw. daß die Bestätigungsnachricht BN2' so behandelt werden kann, als würde sie Daten eines Objektes der Klasse A enthalten. Obwohl das Programm im Bedienrechner 24 nach der Weiterentwicklung im Steuerrechner 36 nicht verändert worden ist, können Objekte der Klasse A' im Steuerrechner 36 vom Steuerrechner 24 aus so geführt werden, als wären sie Objekte der Klasse A. Solange im Bedienrechner A nur die Klasse A bekannt ist, werden in der Bestätigungsnachricht BN2' auch nur die in Objekten der Klasse A enthaltenen Daten bearbeitet.

15

Aufgrund des an Hand der Figuren 3 und 4 erläuterten Verfahrens ist es auch möglich, Klassen zu unterstützen, die zu mehreren Klassen allomorph sind. Wird z.B. die Klasse A' zu einer Klasse A'' weiterentwickelt, so können Objekte zu den Klassen A' und A allomorph sein. Der Steuerrechner 36 kann dann von Bedienrechnern 24 ausgeführt werden, bei deren Programmierung mindestens eine der Klassen A, A' oder A'' bekannt war. Auf eine Unterstützung der Klasse A kann verzichtet werden, sobald alle Bedienrechner zumindest die Klasse A'

25 kennen.

30

Das an Hand der Figuren 3 und 4 erläuterte Verfahren wird bei Wartungsnachrichten zum Erzeugen und bei Wartungsnachrichten zum Lesen von Objekten angewendet. Außerdem können mit diesem Verfahren Daten in Objekten verändert sowie Objekte gelöscht werden. Damit diese Verfahren fehlerfrei arbeiten, muß der Bedienrechner 24 fähig sein:

Die Bestätigungsnachrichten BN den zugehörigen Wartungs nachrichten WN zuzuordnen,

- das Kennzeichen allo in den Bestätigungsnachrichten BN zu lesen und auszuwerten,
- unbekannte Namensbindungen (vgl. Erläuterungen zu Fig. 5)

 zu übergehen und unbekannte Kennzeichenwerte und Parameter zu übergehen,
 - falls im Bedienrechner eine Datenbank vorhanden ist die Ursprungsklasse eines Objekts zu speichern,

- unbekannte wahlfreie Werte zu übergehen, und
- unbekannte Werte vom Datentyp "enumerated" zu übergehen.
- 15 Mit anderen Worten muß der Bedienrechner 24 so programmiert sein, daß von ihm aus Objekte in einen Steuerrechner mit größerem Wissen geführt werden können.

Das Schnittstellenprogramm 100 hat folgende Merkmale:

20

25

- Beim Erweitern einer Klasse ist es erforderlich, die erweiterte Klasse in die Tabelle T aufzunehmen, das bedeutet, daß für jede erweiterte Klasse die Klassen gespeichert werden müssen, zu denen die erweiterte Klasse
 allomorph ist.
- Die Namensbindungen, welche von der erweiterten Klasse betroffen werden, müssen in der Tabelle T gespeichert werden.

30

35

- Das Schnittstellenprogramm 100 soll aus den vom Anwendungsprogramm 102 kommenden Bestätigungsnachrichten BN die Daten entfernen, die nicht zur kompatiblen Klasse gehören, falls die an den Bedienrechner 24 gesendete Bestätigungsnachricht BN die nicht erweiterte Klasse betrifft.

20

25

30

- Das Schnittstellenprogramm 100 muß Filterbefehle bearbeiten können. Dies wird unten an Hand der Figur 5 noch genauer erläutert.
- 5 Das Anwendungsprogramm 102 erfüllt die folgenden Anforderungen:
- Sobald zur erweiterten Klasse übergegangen wird, werden unabhängig vom Wissen des Bedienrechners nur noch Objekte der neuen Klasse erzeugt.
 - Die neuen Klassen sind entweder erweiterte Klassen oder Klassen, die mit den bisherigen Klassen nichts zu tun haben.

Das Datenbankprogramm 110 ist so ausgelegt, daß nach einer Weiterentwicklung der gesamte Datenbestand zur Ursprungsklasse in einen Datenbestand der erweiterten Klasse umgewandelt wird.

Das Erstellen der Tabelle T und das Umwandeln des Datenbestandes in der Datenbank des Datenbankprogramms 110 werden durch Hilfsprogramme unterstützt. Diese Hilfsprogramme werten Beschreibungssprachen aus, mit deren Hilfe die Erweiterung von Klassen angegeben wird.

Im folgenden werden Bedingungen angesprochen, die bei der Berücksichtigung von Allomorphie eingehalten werden müssen. Diese Bedingungen gelten auf der Ebene der Klassen, obwohl gemäß Standard X.720 Allomorphie zunächst eine Eigenschaft eines Objekts ist. Im Standard X.720 sind folgende Bedingungen angesprochen:

- Bedingungen für die erweiterte Klasse in Abschnitt 5.2.2.1,

- Bedingungen für sogenannte Programmpakete in den Abschnitten 5.2.2.1 und 5.2.2.2,
- Bedingungen für Kennzeichenwerte in den Abschnitten 5.2.2.3 und 5.3.4.1,
 - Bedingungen für sogenannte Kennzeichengruppen in Abschnitt 5.2.2.4,
- 10 Bedingungen für Aktionen, Bestätigungen und Parameter in Abschnitt 5.2.2.54,
 - Bedingungen für das Verhalten der Objekte in Abschnitt 5.2.2.6, und
 - Bedingungen für die Namensbindung in Abschnitt 5.3.4.1.

Figur 5 zeigt in einem Teil a einen Ausschnitt aus einem sogenannten binären Baum, der die Namensbindung von Objekten 20 der Klasse A und A' festlegt. Namensbindung ist die für das Festlegen eines eindeutigen Namens für ein Objekt verwendete Zuordnung des Objektes zu einem sogenannten übergeordneten Objekt. Gleiche Objektnamen für unterschiedliche Objekte können verwendet werden, wenn die Objekte jeweils zu einem ande-25 ren übergeordneten Objekt gehören. Aus dem Namen des übergeordneten Objekts und dem Namen der so untergeordneten Objekte entsteht dann ein im Steuerrechner 16 eindeutiger Name. Das übergeordnete Objekt ist beim Erzeugen eines Objektes anzugeben. Das untergeordnete Objekt wird als im übergeordneten Ob-30 jekt enthalten angesehen, englisch als "Containment" bezeichnet.

Ein Objekt bl der Klasse B ist das übergeordnete Objekt für die Objekte a2 und a3, die beide nach der Weiterentwicklung zur Klasse A' gehören, jedoch unterschiedliche ursprüngliche Klassen oC=A bzw. oC=A' haben. In der Tabelle T, die vom Schnittstellenprogramm 100 verwendet wird, sind die Namens-

bindungen vermerkt. Für den im Teil A der Figur 5 gezeigten Ausschnitt des Namensbaumes ist in der Tabelle T vermerkt, daß Objekte der Klasse B untergeordnete Objekte der ursprünglichen Klasse A bzw. der Klasse A' enthalten. Bei jeder Weiterentwicklung werden auch die Namensbindungen in der Tabelle T an den Namensbaum angepaßt, der nach der Weiterentwicklung maßgeblich ist.

Im Teil b der Figur 5 ist ein Zugriff auf Objekte beider

Klassen A und A' dargestellt, nachdem auch der Bedienrechner

24 beide Klassen A und A' kennt. Eine Wartungsnachricht WN3

enthält eine Filteranweisung Fl=((oC=A)ODER(oC=A')), in der

festgelegt ist, daß untergeordnete Objekte der Klasse A oder

A' erfaßt werden sollen. Das Klassenkennzeichen moC=B der

Wartungsnachricht WN3 gibt an, daß sich die Wartungsnachricht

WN3 auf die Klasse B bezieht. Das Objektkennzeichen moI=bl

der Wartungsnachricht WN3 gibt an, daß das Objekt b1 bearbei
tet werden soll, d.h. das übergeordnete Objekt ist.

- Beim Ausführen des Schnittstellenprogramms 100 in der Vermittlungsstelle 16 wird die Filteranweisung F1 unverändert an das Anwendungsprogramm 102 gesendet. Das bedeutet insbesondere, daß in der Filteranleitung F1 Operationen, die sich auf die ursprüngliche Klasse A beziehen, nicht durch Operationen ersetzt werden, die sich auf die erweiterte Klasse A' beziehen. Diese Maßnahme gewährleistet, daß der Bedienrechner 24 zwischen Objekten der ursprünglichen Klasse A und der ursprünglichen Klasse A' unterscheiden kann.
- Das Anwendungsprogramm 102 veranlaßt, daß mittels Datenbankprogramm 110 aus dem Speicher 42 die Daten der Objekte a2 und
 a3 gelesen werden, die als ursprüngliche Klasse die Klasse A
 bzw. A' haben. An das Schnittstellenprogramm 100 werden zwei
 nicht dargestellte Bestätigungsnachrichten BN3 und BN4 gesendet. Im Schnittstellenprogramm 100 wird aus der Bestätigungsnachricht BN3 eine neue Bestätigungsnachricht BN3' erzeugt,
 die das Klassenzeichen moC=A, das Objektkennzeichen moI=a2,

das Ursprungskennzeichen oC=A und das Hilfskennzeichen allo ={} enthält. Aus der Bestätigungsnachricht BN4 wird im Schnittstellenprogramm 100 eine Bestätigungsnachricht BN4' erzeugt, welche das Klassenkennzeichen moC=A', das Objekt-kennzeichen moI=a3, das Ursprungskennzeichen oC=A', und das Hilfskennzeichen allo={A} enthält.

Der CCITT-Standard X.734 "Information Technology - Open Systems Interconnection - Systems Management: Event Report Ma-10 nagement Function" aus dem Jahre 1993 erläutert die Ereignissteuerung im Führungsnetz 10, vgl. Figur 1. Es werden sogenannte Diskriminatoren verwendet, die Ereignisse innerhalb der Vermittlungsstelle 16 nur unter bestimmten Bedingungen an den Bedienrechner 24 weiterleiten. Nach der Weiterentwicklung 15 der Klasse A zur Klasse A' genügt es, die Diskriminatoren, welche die Klasse A betreffen, in Diskriminatoren umzuwandeln, welche die Klasse A' betreffen. Werden nach der Weiterentwicklung neue Diskriminatoren erzeugt, so wird die Klasse A durch die Klasse A' ersetzt, wenn die Klasse A als Auswahl-20 kriterium für das Weiterleiten der Nachrichten angegeben ist.

Patentansprüche

- 1. Verfahren zum Betreiben eines Telekommunikationsnetzes (10, 12),
- 5 bei dem ein Netzelement (16) an einem Netzknoten eines Telekommunikationsnetzes (12) von einem Steuerrechner (36) gesteuert wird,
 - im Steuerrechner (36) neben dem Betriebssystem mehrere Anwendungsprogramme (102, 104) gespeichert sind, bei deren Aus-
- führen Anwendungsobjekte (a2, a3) bearbeitet werden, die Anwendungsobjekte (a2, a3) je nach Zugehörigkeit zu einer Klasse (A, A') Daten mit einer vorgegebenen Datenstruktur sowie vorzugsweise auch vorgegebene Verfahren zum Bearbeiten der Daten haben,
- zwischen einem Bedienrechner (24) und dem Steuerrechner (36) eine Verbindung aufgebaut wird, über die der Steuerrechner (36) mittels mindestens einer Wartungsnachricht (WN1 bis WN3) gewartet wird,
 - die Wartungsnachricht (WN1 bis WN3) ein Klassenkennzeichen
- 20 (moC) enthält, das die Wartungsnachricht (WN1 bis WN3) einer Klasse (A, A') zuordnet,
 - das Klassenkennzeichen (moC) der Wartungsnachricht (WN1 bis WN3) die im Bedienrechner bekannte Klasse (A) eines zu bearbeitenden Anwendungsobjektes (a2, a3) angibt,
- beim Abarbeiten des Schnittstellenprogramms (100) anhand des Klassenkennzeichens (moC) ein Ersatzkennzeichen ermittelt wird, welches eine Ersatzklasse (A') angibt, der das zu bearbeitende Anwendungsobjekt (a2, a3) im Netzelement (16) angehört,
- 30 beim Abarbeiten des Schnittstellenprogramms (100) das Ersatzkennzeichen (A') in eine geänderte Wartungsnachricht (WN1' bis WN2') aufgenommen wird,
 - und bei dem beim Bearbeiten der geänderten Wartungsnachricht (WN1' bis WN2') durch ein Anwendungsprogramm (102) das zu be-
- arbeitende Anwendungsobjekt (a2, a3) als Objekt der Ersatzklasse (A') bearbeitet wird.

- 2. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß beim Ermitteln des Ersatzkennzeichens (A') eine im Speicher (122) des Steuerrechners
 (16) gespeicherte erste Tabelle (T) verwendet wird, in der
 dem Klassenkennzeichen (A) ein Ersatzkennzeichen (A') zugeordnet ist.
- 3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß das Anwendungsprogramm (102) nach dem Bear10 beiten der geänderten Wartungsnachricht (WN1') eine Bestätigungsnachricht (BN1) erzeugt, in der die beim Erzeugen des zu
 bearbeitenden Anwendungsobjekts (a2, a3) angegebene Klasse
 (oC) als Klassenkennzeichen (moC) angegeben ist.
- 15 4. Verfahren nach Anspruch 3, dadurch gekennzeichnet, daß beim Abarbeiten des Schnittstellenprogramms (100)
 aus der Bestätigungsnachricht (BN1) eine geänderte Bestätigungsnachricht (BN1') erzeugt wird, die nur solche Daten enthält, die ein Anwendungsobjekt (a2) der Klasse (A) hat, auf
 20 die sich die Bestätigungsnachricht (BN1) bezieht,
 und daß die geänderte Bestätigungsnachricht (BN1') an den Bedienrechner (24) gesendet wird.
- 5. Verfahren nach Anspruch 3 oder 4, dadurch gekenn25 zeichnet, daß die beim Erzeugen des zu bearbeitenden Anwendungsobjektes (a2, a3) angegebene Klasse (A, A') als Ursprungsklasse (oC) in den Daten das zu bearbeitenden Anwendungsobjekts (a2, a3) gespeichert ist,
 und daß beim Abarbeiten des Anwendungsprogramms (102) die Ursprungsklasse (oC) als Klassenkennzeichen (moC) verwendet
 wird.
- 6. Verfahren nach einem der Ansprüche 3 bis 5, dadurch gekennzeichnet, daß die Bestätigungsnachricht (BN1)
 35 ein Hilfskennzeichen (allo) enthält, in welchem mindestens eine Klasse (A) bezeichnet ist, die im Bedienrechner (24) und/oder in mindestens einem anderen Bedienrechner als die

Klasse (A) bekannt ist, zu der das zu bearbeitende Anwendungsobjekt (a2, a3) gehört, vorzugsweise zumindest die im Bedienrechner (24) bekannte Klasse (A) des zu bearbeitenden Anwendungsobjekts (a2, a3).

5

7. Verfahren nach Anspruch 6, dadurch gekennzeichnet, daß die Bestätigungsnachricht (BN1) neben dem Klassenkennzeichen (moC) ein Ursprungskennzeichen (oC) enthält, in welchem die Ursprungsklasse (A, A') angegeben ist.

10

- 8. Verfahren nach Anspruch 6 oder 7, dadurch gekennzeichnet, daß mindestens eine im Bedienrechner (24)
 und/oder in mindestens einem anderen Bedienrechner für das
 Anwendungsobjekt bekannte Klasse (A) als Allomorphklasse (al-
- 15 lo) in den Daten des Anwendungsobjektes (a2, a3) gespeichert ist,

und daß beim Abarbeiten des Anwendungsprogramms (102) die Allomorphklasse (allo) als Hilfskennzeichen (allo) verwendet wird.

20

- 9. Verfahren nach einem der Ansprüche 4 bis 8, dadurch gekennzeichnet, daß die bzw. eine beim Abarbeiten des Schnittstellenprogramms (100) für den Bedienrechner (24) erzeugte Bestätigungsnachricht (BN1') das Hilfskennzeichen (al-
- lo) und/oder das Klassenkennzeichen (moC) und/oder das Ursprungskennzeichen (oC) enthält.
 - 10. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß das Netzelement eine Vermittlungsstelle (16), ein Cross-Connector oder eine Konzentratoreinheit ist.
- 11. Verfahren nach einem der vorhergehenden Ansprüche, da-durch gekennzeichnet, daß das Telekommunikationsnetz
 (12) ein Festnetz, ein Mobilfunknetz oder ein Netz mit einem Festnetzanteil und einem Mobilfunknetzanteil ist.

12. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß das Schnittstellenprogramm
(100) weitere Schnittstellenfunktionen zwischen dem Bedienrechner (24) und den Anwendungsprogrammen (102, 104) ausführt,

vorzugsweise eine Ereignissteuerung zum Festlegen der Bearbeitungsreihenfolge der Wartungsnachrichten (WN1 bis WN3) und/oder eine Anpassung der vom Bedienrechner (24) kommenden Nachrichten (WN1 bis WN3) an ein Protokoll zur Nachrichten-

übertragung innerhalb des Steuerrechners (16), und/oder eine Anpassung der von den Anwendungsprogrammen (102, 104) kommenden Bestätigungsnachrichten (BN1, BN2) an ein vorgegebenes Protokoll zur Nachrichtenübertragung zwischen Bedienrechner (24) und Steuerrechner (16).

15

5

13. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß ein erstes Anwendungsprogramm (102) zur Teilnehmerverwaltung,

und/oder ein zweites Anwendungsprogramm zum Verwalten von

- Verbindungsleitungen zu anderen Vermittlungseinrichtungen, und/oder ein drittes Anwendungsprogramm zur Wartung der Vermittlungseinrichtungen,
 - und/oder ein viertes Anwendungsprogramm (104) zur Verkehrsmessung der geschalteten Verbindungen (18) verwendet wird.

25

30

- 14. Verfahren nach Anspruch 13, dadurch gekennzeichnet, daß die Anwendungsobjekte (a2, a3) des ersten Anwendungsprogramms (102) für jeweils einen Teilnehmer (Tln2,
 Tln3) die Teilnehmerdaten enthalten, vorzugsweise die Rufnummer und/oder die nutzbaren Telekommunikationsdienste.
- 15. Verfahren nach einem der vorhergehenden Ansprüche, da-durch gekennzeichnet, daß die Wartungsnachrichten (WN1 bis WN3) ein Namenskennzeichen (moI) für den Namen des Anwendungsobjektes (a2, a3) enthalten, auf welches sich die Wartungsnachricht (WN1 bis WN3) bezieht.

- 16. Netzelement (16) zum Betreiben eines Telekommunikationsnetzes (10, 12), dadurch gekennzeichnet, daß es einen Speicher zum Speichern von Befehlen hat, bei deren Abarbeiten das Verfahren nach einem der vorhergehenden Ansprüche
 durchgeführt wird.
 - 17. Telekommunikationsnetz (10, 12), gekennzeichnet durch, ein Netzelement gemäß Anspruch 16.

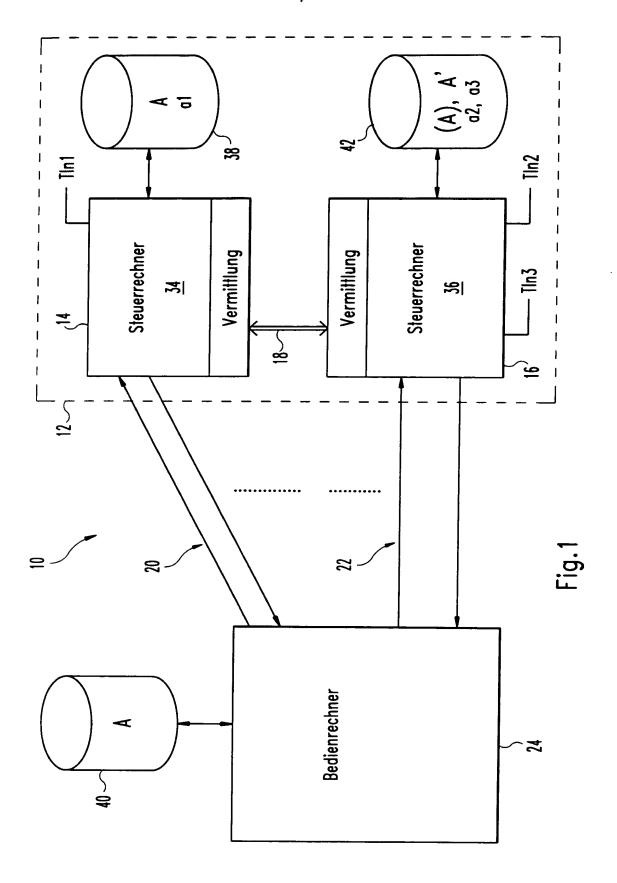
Zusammenfassung

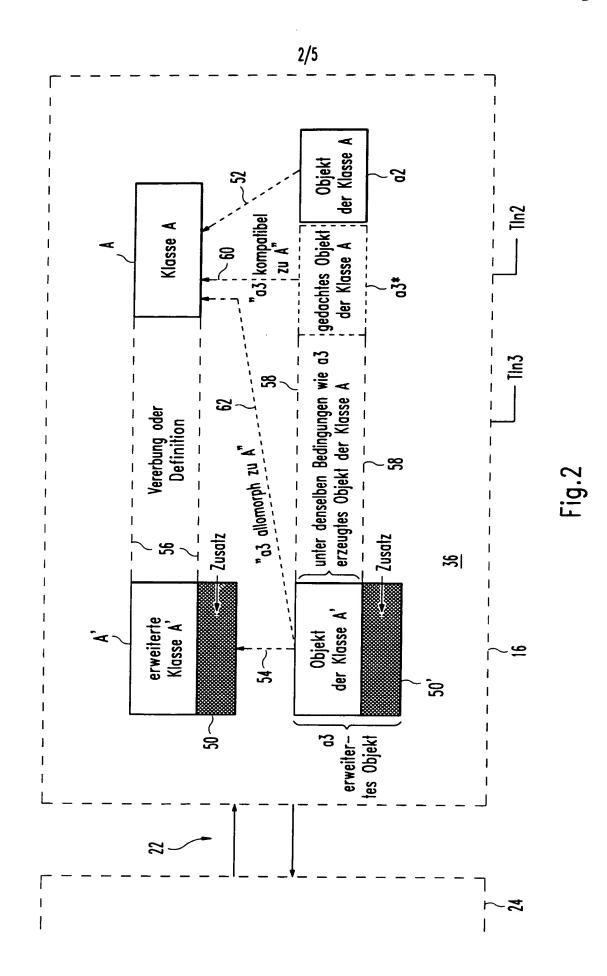
Verfahren und Netzelement zum Betreiben eines Telekommunikationsnetzes

5

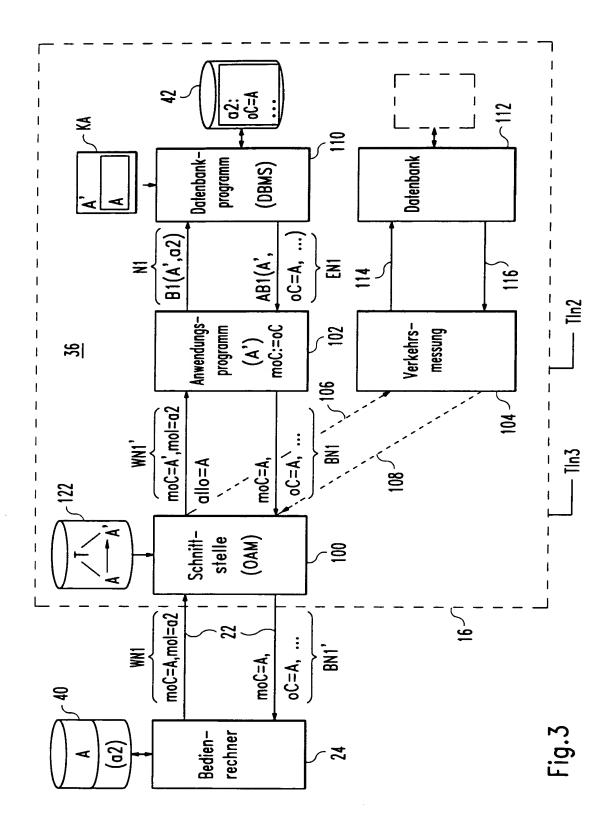
Erläutert wird ein Verfahren zum Betreiben eines Telekommunikationsnetzes. Ein Netzelement (16) an einem Netzknoten des
Telekommunikationsnetzes wird von einem Steuerrechner (36)
gesteuert. Der Steuerrechner (16) wird von einem Bedienrechner (24) aus gewartet. Beim Weiterentwickeln von Anwendungsprogrammen (102, 104) wird Allomorphie berücksichtigt, damit
auch ein nicht weiterentwickelter Bedienrechner (24) den
Steuerrechner (36) warten kann. Der Aufwand beim Berücksichtigen von Allomorphie im Steuerrechner (36) ist klein, weil
ein Schnittstellenprogramm (100) verwendet wird, in dem wesentliche Bearbeitungsschritte für alle Anwendungsprogramme
(102,104) durchgeführt werden, die das Berücksichtigen von
Allomorphie gewährleisten.

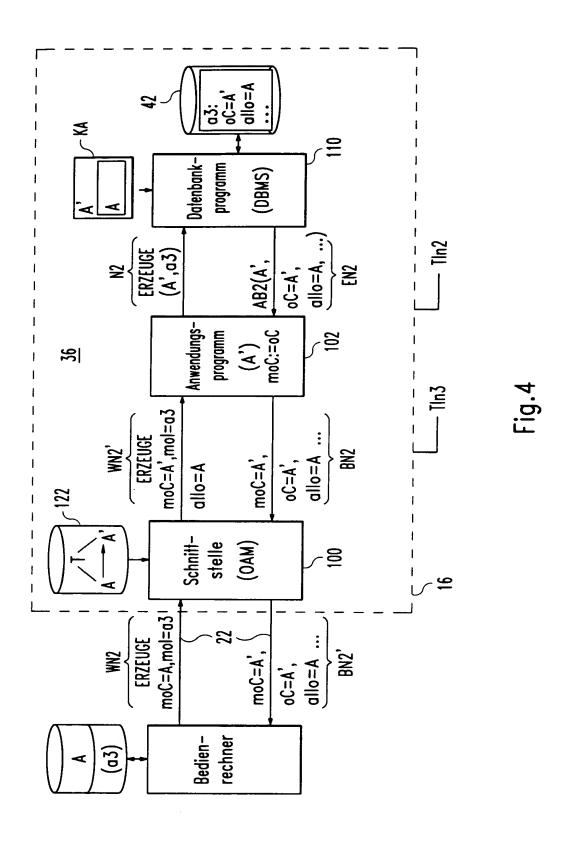
20 Figur 3





MOS SILL COM.





ing a second of the

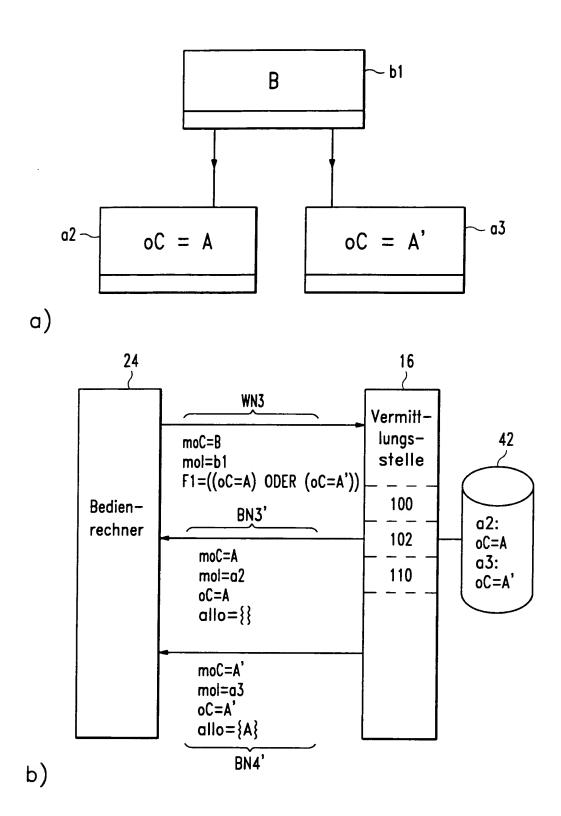


Fig.5

WELTORGANISATION FÜR GEISTIGES EIGENTUM

INTERNATIONALE ANMELDUNG VERÖFFENTLICHT NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT)

(51) Internationale Patentklassifikation 7:

H04Q 3/00

(11) Internationale Veröffentlichungsnummer: A1

WO 00/54520

(43) Internationales

Veröffentlichungsdatum:

NL, PT, SE).

14. September 2000 (14.09.00)

(21) Internationales Aktenzeichen:

PCT/EP00/02070

(22) Internationales Anmeldedatum:

9. März 2000 (09.03.00)

(30) Prioritätsdaten:

99104749.9

10. März 1999 (10.03.99)

EP

Veröffentlicht

Mit internationalem Recherchenbericht.

(81) Bestimmungsstaaten: CN, US, europäisches Patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC,

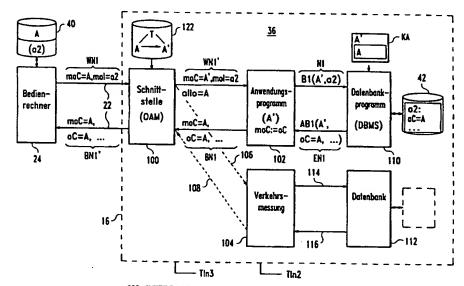
- (71) Anmelder (für alle Bestimmungsstaaten ausser US): SIEMENS AKTIENGESELLSCHAFT [DE/DE]; Wittelsbacherplatz 2, D-80333 München (DE).
- (72) Erfinder; und
- (75) Erfinder/Anmelder (nur für US): NEUHAUS, Matthias [DE/DE]; Trivastr. 15C, D-80637 München (DE). FRÖSCHL, Maria [AT/DE]; Therese-Giese-Allee 79, D-81739 München (DE).
- (74) Gemeinsamer Vertreter: **SIEMENS** AKTIENGE-SELLSCHAFT; Postfach 22 16 34, D-80506 Munchen (DE).
- (54) Title: METHOD AND NETWORK ELEMENT FOR OPERATING A TELECOMMUNICATIONS NETWORK
- (54) Bezeichnung: VERFAHREN UND NETZELEMENT ZUM BETREIBEN EINES TELEKOMMUNIKATIONSNETZES

(57) Abstract

The invention relates to a method for operating a telecommunications network. A network element (16) situated on a node of the telecommunications network is controlled by a control computer (36). Said control computer (36) is controlled by an operator computer (24). Allomorphy is taken into consideration in the improvement of application programs (102, 104) so that even a non-improved operator computer (24) is able to control the control computer (36). Making allowance for allomorphy results in little additional complexity since an interface program (100) is provided for in which essential processing steps are executed for all application programs (102, 104) and thus ensure that allomorphy is taken into consideration.

(57) Zusammenfassung

Erläutert wird ein Verfahren zum Betreiben eines Telekommunikationsnetzes. Ein Netzelement (16) an einem Netzknoten des Telekommunikationsnetzes wird von



100...INTERFACE 102..APPLICATION PROGRAMME 110...DATA BANK PROGRAMME 112...DATA BANK

104...TRAFFIC MEASUREMENT 24...OPERATOR COMPUTER

einem Steuerrechner (36) gesteuert. Der Steuerrechner (16) wird von einem Bedienrechner (24) aus gewartet. Beim Weiterentwickeln von Anwendungsprogrammen (102, 104) wird Allomorphie berücksichtigt, damit auch ein nicht weiterentwickelter Bedienrechner (24) den Steuerrechner (36) warten kann. Der Aufwand beim Berücksichtigen von Allomorphie im Steuerrechner (36) ist klein, weil ein Schnittstellenprogramm (100) verwendet wird, in dem wesentliche Bearbeitungsschritte für alle Anwendungsprogramme (102, 104) durchgeführt werden, die das Berücksichtigen von Allomorphie gewährleisten.

bau solcher offenen Systeme. Zum Führen des Tk-Netzes soll ein separates Führungsnetz verwendet werden. Die Schnittstellen zwischen Bedienrechner und Vermittlungseinrichtung sind in Protokollen Q1, Q2 und Q3 standardisiert.

5

7:A

Die Anwendungsobjekte sind als Objekte einer objektorientierten Sprache definiert, z.B. in der Sprache C++ oder CHILL.
Werden die Anwendungsprogramme weiterentwickelt, so muß gewährleistet werden, daß das Führungsnetz auch mit den neuen Anwendungsprogrammen fehlerfrei arbeitet. Das bedeutet insbesondere, daß Anwendungsobjekte, die vom Bedienrechner als zu einer ursprünglichen Klasse gehörend angesehen werden, nicht ohne weiteres einer geänderten Ersatzklasse zugeordnet werden können.

15

20

25

Dieses Problem wird im CCITT-Standard X.720 (01/92) - "Information Technology - Open Systems Interconnection - Structure of Management Information: Management Information Model" - im Abschnitt 5.2.1 angesprochen. Im Abschnitt 5.2.3 des Standards X.720 werden zwei Methoden zum Lösen des Problems vorgegeben. Bei der ersten Methode wird auf der Seite des Anwendungsprogramms eine Programmiertechnik verwendet, die Allomorphie berücksichtigt. Allomorphie ist die Fähigkeit eines bestimmten Anwendungsobjektes der Ersatzklasse so geführt zu werden, als wäre es ein Objekt der ursprünglichen Klasse, wenn diese Fähigkeit durch Maßnahmen auf der Seite des Anwendungsprogramms entsteht. Die andere Methode besteht darin, daß auf der Seite des Bedienrechners Maßnahmen getroffen werden, welche auch bei einer Weiterentwicklung des Anwendungsprogramms ein Zusammenarbeiten zwischen Bedienrechner und Anwendungsprogramm ermöglichen.

30

35

Es ist Aufgabe der Erfindung zum Betreiben eines Telekommunikationsnetzes ein einfaches Verfahren anzugeben, bei dem Allomorphie berücksichtigt wird. Beschreibung

Verfahren und Netzelement zum Betreiben eines Telekommunikationsnetzes

5

Die Erfindung betrifft ein Verfahren zum Betreiben eines Telekommunikationsnetzes, kurz Tk-Netz, bei dem ein Netzelement an einem Netzknoten des Tk-Netzes von einem Steuerrechner gesteuert wird. Das, Netzelement ist beispielsweise eine Ver-10 mittlungsstelle zum Vermitteln von Verbindungen, ein sogenannter Cross-Connector oder eine Konzentratoreinheit zum Anschluß mehrerer Teilnehmer an eine Verbindungsleitung. Im Steuerrechner sind neben dem Betriebssystem zum Betreiben des Steuerrechners mehrere Anwendungsprogramme gespeichert, bei deren Ausführen Anwendungsobjekte bearbeitet werden. Zu den 15 Anwendungsobjekten gehören Daten mit einer vorgegebenen Datenstruktur sowie vorzugsweise auch vorgegebene Verfahren zum Bearbeiten der Daten. Die Datenstruktur und die Verfahren sind abhängig von einer auch beim Erzeugen des jeweiligen Anwendungsobjektes anzugebenden Klasse. Zwischen einem Be-20 dienrechner und einem Steuerrechner wird eine Verbindung aufgebaut, über die der Steuerrechner mittels Wartungsnachrichten gewartet wird.

Derartige Verfahren werden zum Führen des Tk-Netzes eingesetzt, wenn zum Beispiel als Netzelement eine neue Vermittlungseinrichtung in Betrieb genommen wird oder wenn später Teilnehmerdaten in der Vermittlungseinrichtung geändert werden müssen, wie es beim Anschluß neuer Teilnehmer oder beim Umzug eines bisherigen Teilnehmers der Fall ist. Leistungsfähige Verfahren zum Führen des Tk-Netzes entstehen, wenn sogenannte offene Systeme verwendet werden, bei deren Programmierung weltweit geltende Standards beachtet werden. Beispielsweise betreffen Standards der ISO (International Standardisation Organization) und der ITU (International Telecommunication Union) mit ihrem Organ ITU-T, früher CCITT (International Telegraph and Telephone Consultativ Committee), den Auf-

35

Diese Aufgabe wird durch ein Verfahren mit den Merkmalen des Patentanspruchs 1 gelöst. Weiterbildungen sind in den Unteransprüchen angegeben.

Beim erfindungsgemäßen Verfahren wird aus den Wartungsnachrichten beim Abarbeiten eines für mehrere Anwendungsprogramme genutzten Schnittstellenprogramms jeweils ein Klassenkennzeichen ermittelt, in welchem die Klasse angegeben ist, auf die sich die Wartungsnachricht bezieht. Das Klassenkennzeichen in 10 der Wartungsnachgicht gibt die im Bedienrechner bekannte Klasse eines zu Bearbeitenden Anwendungsobjektes an. Durch die Weiterentwicklung kommt es vor, daß die im Bedienrechner bekannte Klasse von der tatsächlichen Klasse des Anwendungsobjekts abweicht. Beim Abarbeiten des Schnittstellenprogramms wird an Hand des Klassenkennzeichens ein Ersatzkennzeichen 15 ermittelt, welches eine Ersatzklasse angibt, der das Anwendungsobjekt im Netzelement zugeordnet ist. Das Ersatzkennzeichen wird in eine geänderte Wartungsnachricht aufgenommen. Beim Bearbeiten der geänderten Wartungsnachricht durch ein 20 Anwendungsprogramm wird das Anwendungsobjekt dann als zur Ersatzklasse gehörend bearbeitet. Dies ist möglich, weil das Anwendungsobjekt allomorph zu der im Bedienrechner bekannten Klasse ist, die in der ungeänderten Wartungsnachricht vom Bedienrechner für das Anwendungsobjekt vorausgesetzt worden 25 ist.

Das Schnittstellenprogramm übernimmt zentral für mehrere Anwendungsprogramme die Zuordnung der Ersatzkennzeichen zu den
Klassenkennzeichen. Beim erfindungsgemäßen Verfahren muß dieser Schritt nicht in jedem Anwendungsprogramm sondern nur
einmal im Schnittstellenprogramm programmiert werden. Bei
mehreren hundert Anwendungsprogrammen je Steuerrechner verringert sich dadurch der Programmier-, Wartungs- und Dokumentationsaufwand erheblich. Die Anwendungsprogamme werden von
zusätzlichen Schritten freigehalten, die beim Berücksichtigen
von Allomorphie notwendig sind, weil diese Schritte zentral
im vorgelagerten Schnittstellenprogramm durchgeführt werden.

tert.

30

35

Ein Teil der zusätzlichen Schritte wird auch in nachgelagerten Datenbanken durchgeführt, welche von den Anwendungsprogrammen genutzt werden.

Das Ausführen der Zuordnung von Ersatzkennzeichen und Hilfskennzeichen in einem zentralen Schnittstellenprogramm ist möglich, weil Allomorphie beim erfindungsgemäßen Verfahren auf Klassenebene definiert ist. Eine derartige Definition ist im Standard X.720 nicht erwähnt aber dennoch standardgerecht.

Allomorphie auf Klassenebene bedeutet, daß alle Objekte der Ersatzklasse so geführt werden können, als wären sie Objekte der ursprünglichen Klasse. Durch eine auf alle Objekte der Ersatzklasse bezogene Definition von Allomorphie entstehen dann keine Nachteile, wenn vorgegebene Programmierregeln beachtet werden. Beispiele für solche Programmierregeln werden unten im Zusammenhang mit den Ausführungsbeispielen erläu-

Das erfindungsgemäße Verfahren ermöglicht es, die Vorgaben

des Standards X.720 auf eine einfache Art und Weise einzuhalten. Die Anwendungsprogramme im Steuerrechner können mit einem geringen Mehraufwand weiterentwickelt werden, wobei immer sichergestellt bleibt, daß auch bei unverändert gebliebenen Programmen im Bedienrechner keine Fehler beim Betreiben des

Führungsnetzes auftreten.

In einer Weiterbildung wird im Schnittstellenprogramm eine Tabelle verwendet, mit der dem Klassenkennzeichen die Ersatz-kennzeichen zugeordnet werden. Die Tabelle ist im Speicher des Steuerrechners gespeichert. Ein Eintrag der Tabelle wird gelesen, indem eine dem Klassenkennzeichen zugeordnete Speicherzelle gelesen wird, die das zum Klassenkennzeichen gehörende Ersatzkennzeichen enthält. Das Ermitteln des Ersatzkennzeichens benötigt so nur einen einzigen Lesezugriff auf den Speicher. Ändern sich durch Weiterentwicklungen der Anwendungsprogramme die Ersatzkennzeichen, so müssen nur die Speicherinhalte neu programmiert werden. Das bedeutet, daß

der Inhalt der Tabelle leicht ausgetauscht oder erweitert werden kann.

Wird in einer anderen Weiterbildung durch das Anwendungsprogramm nach dem Bearbeiten der geänderten Wartungsnachricht eine Bestätigungsnachricht erzeugt, in der die beim Erzeugen des zu bearbeitenden Anwendungsobjektes angegebene Klasse als Klassenzeichen angegeben ist, so kann die Bestätigungsnachricht vom Schnittstellenprogramm nachfolgend auf einfache Art 10 weiter bearbeitet werden. Beispielsweise kann beim Abarbeiten des Schnittstel enprogramms an Hand des Klassenkennzeichens festgestellt werden, welche Daten aus der Bestätigungsnachricht zu entfernen sind. Dazu wird die im Schnittstellenprogramm verwendete Tabelle so erweitert, daß zu jedem Klassenkennzeichen auch Eintragungen zu den erlaubten Daten gehören. 15 Das Schnittstellenprogramm erzeugt dann aus der Bestätigungsnachricht eine neue Bestätigungsnachricht, die nur solche Daten eines Anwendungsobjekts der Klasse enthält, auf die sich die Bestätigungsnachricht bezieht.

20

25

1

Die beim Erzeugen des zu bearbeitenden Anwendungsobjektes angegebene Klasse wird in einer Weiterbildung als Ursprungsklasse in den Daten des zu bearbeitenden Anwendungsobjekts gespeichert. Beim Abarbeiten des Anwendungsprogramms wird die Ursprungsklasse dann als Klassenkennzeichen verwendet. Durch diese Vorgehensweise ist die Ursprungsklasse auf einfache Art und Weise verfügbar.

Enthält die Bestätigungsnachricht in einer anderen Weiterbildung auch ein Hilfskennzeichen, in welchem mindestens eine
Klasse bezeichnet ist, die im Bedienrechner und/oder in mindestens einem anderen Bedienrechner als die Klasse bekannt
ist, zu der das zu bearbeitende Anwendungsobjekt gehört, so
kann später das Programm im Bedienrechner an Hand des Hilfskennzeichens ermitteln, wie die empfangene Bestätigungsnachricht zu bearbeiten ist. Dies ist insbesondere dann von Bedeutung, wenn die im Klassenkennzeichen der vom Bedienrechner

empfangenen Bestätigungsnachricht angegebene Klasse im Bedienrechner noch nicht bekannt ist. Der Bedienrechner bestimmt die Klasse, auf die sich die Bestätigungsnachricht bezieht dann an Hand der im Hilfskennzeichen angegebenen Klasse bzw. Klassen. Das Hilfskennzeichen enthält, mit anderen Worten ausgedrückt, die Klassen, zu denen das Anwendungsobjekt allomorph ist. Enthält die Bestätigungsnachricht neben dem Klassenkennzeichen auch ein Ursprungskennzeichen, in dem die Ursprungsklasse angegeben ist, so lassen sich die Vorgaben des Protokolls für den Nachrichtenaustausch im Steuerrechner und auch für das Protokoll für den Nachrichtenaustausch zwischen Bedienrechner und Steuerrechner erfüllen.

In einer anderen Weiterbildung wird mindestens eine im Bedienrechner und/oder in mindestens einem anderen Bedienrechner für das Anwendungsobjekt bekannte Klasse als Allomorphklasse in den Daten des Anwendungsobjekts gespeichert. Beim
Abarbeiten des Anwendungsprogramms wird dann die Allomorphklasse als Hilfskennzeichen verwendet. Durch diese Maßnahme
entsteht eine übersichtliche Datenstruktur, bei der die Anwendungsobjekte ihre Allomorphklassen selbst verwalten. Im
Schnittstellenprogramm und im Anwendungsprogramm müssen keine
zusätzlichen Maßnahmen hinsichtlich der Allomorphklasse getroffen werden.

25

30

10

In einer anderen Weiterbildung ist das Schnittstellenprogramm auch für andere Schnittstellenfunktionen zuständig. Beispielsweise für die Ereignissteuerung zum Festlegen der Bearbeitungsreihenfolge der Wartungsnachrichten oder für Protokollanpassungen dieser Nachrichten, englisch als "basic encoding" bezeichnet. Durch diese Maßnahme gibt es auf dem Steuerrechner nur ein einziges Schnittstellenprogramm, das einheitlich programmiert und gewartet wird.

35 Im folgenden werden Ausführungsbeispiele der Erfindung an Hand der Zeichnungen erläutert. Darin zeigen:

35

- Figur 1 einen Teil eines Führungsnetzes zum Führen eines Telekommunikationsnetzes,
- Figur 2 die Weiterentwicklung einer ursprünglichen Klasse A zu einer erweiterten Klasse A', deren Objekte beim Betreiben des Führungsnetzes wie Objekte der alten Klasse A geführt werden können,
- Figur 3 das Bearbeiten von Nachrichten im Steuerrechner einer Vermittlungseinheit nach der Weiterentwicklung,
 wobei ein Objekt geführt wird, das vor der Weiterentwicklung erzeugt worden ist,
- Figur 4 das Bearbeiten von Nachrichten im Steuerrechner

 nach der Weiterentwicklung, wobei ein Objekt geführt wird, das nach der Weiterentwicklung erzeugt worden ist,
- Figur 5 die Namensbindung der Klassen A und A' sowie ein

 Zugriff auf Objekte der beiden Klassen mittels ein
 ner Filterfunktion.

Figur 1 zeigt einen Teil eines Führungsnetzes 10 zum Führen eines Telekommunikationsnetzes 12, kurz Tk-Netz 12 genannt.

- Das Tk-Netz 12 enthält eine Vielzahl von Vermittlungsstellen, von denen in Figur 1 die Vermittlungsstellen 14 und 16 dargestellt sind. Zum Tk-Netz 12 gehören weiterhin Verbindungsleitungen zwischen den Vermittlungsstellen, von denen in Figur 1 eine Verbindungsleitung 18 zwischen der Vermittlungsstelle 14 und der Vermittlungsstelle 16 dargestellt ist. Das Tk-Netz 12
- und der Vermittlungsstelle 16 dargestellt ist. Das Tk-Netz 12 verbindet die Teilnehmer des Tk-Netzes 12, beispielsweise einen an die Vermittlungsstelle 14 angeschlossenen Teilnehmer Tln1 und einen an die Vermittlungsstelle 16 angeschlossenen Teilnehmer Tln2.

Das Führungsnetz 10 enthält eigene Übertragungsstrecken 20 und 22, die zu einem Bedienrechner 24 führen. Die Übertra-

gungsstrecke 20 überträgt Wartungsnachrichten vom Bedienrechner 24 zur Vermittlungsstelle 14, um beispielsweise in der Vermittlungsstelle 14 Teilnehmerdaten des Teilnehmers Tln1 zu ändern. Die Vermittlungsstelle 14 sendet ihrerseits Bestätigungsnachrichten an den Bedienrechner 24, um die ordnungsgemäße Bearbeitung der empfangenen Wartungsnachricht zu signalisieren. Die Übertragungsstrecke 22 dient zur bidirektionalen Datenübertragung zwischen Bedienrechner 24 und Vermittlungsstelle 16.

10

15

20

Die Wartungsnachbichten werden in der Vermittlungsstelle 14 von einem Steuerrechner 34 und in der Vermittlungsstelle 16 von einem Steuerrechner 36 bearbeitet. Die Datenstrukturen, auf die sich die Wartungsnachrichten beziehen, gehören im Bedienrechner 24 und in der Vermittlungsstelle 14 derselben Klasse A an. Die Vermittlungsstelle 16 enthält dagegen Datenstrukturen einer Klasse A', die gegenüber der im Bedienrechner 24 vorausgesetzten Klasse A weiterentwickelt worden ist. Der fehlerfreie Betrieb des Führungsnetzes 10 wird bezüglich der Vermittlungsstelle 16 dadurch gewährleistet, daß beim Weiterentwickeln der Klasse A zur Klasse A' Allomorphie berücksichtigt worden ist. Was Allomorphie in diesem Zusammenhang bedeutet, wird unten an Hand der Figur 2 erläutert.

Beim in Figur 1 dargestellten Ausführungsbeispiel wird in der Vermittlungsstelle 14 die Klasse A verwendet, die beispielsweise die Datenstruktur für Teilnehmerdaten festlegt, z.B. die Rufnummer und die nutzbaren Dienste des Tk-Netzes 12. Teilnehmerdaten des Teilnehmers Tln1 sind in einem Objekt al gemäß der durch die Klasse A vorgegebenen Datenstruktur in einem Speicher 38 des Steuerrechners 34 gespeichert. Die Klasse A ist auch im Bedienrechner 24 bekannt, angedeutet durch den Buchstaben A in einem Speicher 40 des Bedienrechners 24.

35

In der Vermittlungsstelle 16 wurde die Klasse A zur Klasse A' weiterentwickelt. Ein Objekt a2 enthält beispielsweise die

* * *

Š

Teilnehmerdaten des Teilnehmers Tln2. Das Objekt a2 wurde erstmalig im Speicher 42 gespeichert, bevor die Klasse A zur Klasse A' weiterentwickelt worden ist. Bei der Weiterentwicklung wurde das ursprüngliche Objekt a2 aber umgewandelt, und zwar in ein erweitertes Objekt a2 der Klasse A', indem ein Datenfeld ergänzt worden ist. Ein Objekt a3 im Speicher 42 gehört zur Klasse A' und enthält die Teilnehmerdaten eines Teilnehmers Tln3, dessen Anschluß erst nach der Weiterentwicklung in der Vermittlungsstelle 16 eingerichtet worden

- 10 ist. Obwohl die Programme im Bedienrechner 24 nur Objekte der Klasse A unterstützen, können die zur Klasse A' gehörenden Objekte a2 und a3 vom Bedienrechner 24 aus wie Objekte der Klasse A abgefragt, geändert oder neu eingerichtet werden. Die Erweiterungen der Klasse A' im Vergleich zur Klasse A
- können vom Bedienrechner 24 allerdings erst dann bearbeitet werden, wenn die Programme im Bedienrechner 24 zu einem späteren Zeitpunkt so geändert werden, daß auch die Klasse A' im Bedienrechner 24 bekannt ist.
- Figur 2 zeigt die Klassen A und A' sowie das ursprüngliche Objekt a2 und das Objekt a3. An Hand der Figur 2 wird im folgenden erläutert, was die Bezeichnung "allomorph zu" bedeutet. Die Klasse A' unterscheidet sich von der Klasse A nur durch ein zusätzliches Datenfeld 50. Die Datenstruktur der
- 25 Klasse A wurde also um das Datenfeld 50 erweitert, um eine weitere Eigenschaft der Teilnehmer Tln2, Tln3 beim Betrieb der Vermittlungsstelle 16 berücksichtigen zu können, z.B. ob der Teilnehmer Tln2, Tln3 über einen Lichtwellenleiter oder über einen Kupferleiter an die Vermittlungsstelle 16 ange-
- schlossen ist. Die Klasse A' wird deshalb im folgenden auch als erweiterte Klasse A' bezeichnet. Ein Datenfeld 50' im Objekt a3 enthält als Datum einen Wert, der angibt, daß der Teilnehmer Tln3 mittels eines Lichtwellenleiters an die Vermittlungsstelle 16 angeschlossen ist. Das Objekt a3 wird all-
- 35 gemein als erweitertes Objekt a3 bezeichnet.

Die Klasse, die beim Erzeugen eines Objektes angegeben wird, wird als ursprüngliche Klasse dieses Objekts bezeichnet. Das Objekt a2 hatte als ursprüngliche Klasse die Klasse A, angedeutet durch einen Pfeil 52. Das erweiterte Objekt a3 hat dagegen als ursprüngliche Klasse die erweiterte Klasse A', angedeutet durch einen Pfeil 54.

Eine erste Möglichkeit zum Festlegen der Datenstruktur der Klasse A besteht darin, die Klasse A' aus der Klasse A mit-10 tels sogenannter Vererbung zu erzeugen, die in objektorientierten Programmiersprachen definiert ist. Solche Programmiersprachen sind z.B. die Sprachen C++ und CHILL. Bei einer Vererbung wird durch den Programmierer angegeben, daß die erweiterte Klasse A' von der Klasse A sämtliche Datenstrukturen und sogenannten Methoden zum Bearbeiten der Datenstrukturen übernehmen soll. Weiterhin wird angegeben, daß die Klasse A' zusätzlich das Datenfeld 50 enthält. Eine andere Möglichkeit zum Festlegen der Klasse A' besteht darin, diese Klasse neu zu definieren. In diesem Fall wird die Klasse A' so definiert, wie es bereits bei der Klasse A der Fall war. Zusätz-20 lich wird jedoch noch das Datenfeld 50 definiert. Die Beziehung der übereinstimmenden Teile der Klasse A und der erweiterten Klasse A' ist in Figur 2 mittels Strichlinien 56 dargestellt.

25

In einem gedachten Objekt a3* sind aus dem Objekt a3 alle Datenfelder und alle Methoden zum Bearbeiten der Datenfelder enthalten, die auch beim Einrichten des Teilnehmers Tln3 vor der Weiterentwicklung erzeugt worden wären, als es zwar die Klasse A, aber noch nicht die Klasse A' gab. Im Objekt a3* fehlt deshalb ein dem Datenfeld 50 bzw. 50' entsprechendes Datenfeld. Dieser Sachverhalt wird durch Strichlinien 58 verdeutlicht. Das Objekt a3* ist eine Anschauungshilfe zum Abgrenzen der Begriffe "kompatibel zu" und "allomorph zu". Ein Pfeil 60 verdeutlicht, daß das Objekt a3* kompatibel zur Klasse A ist, weil es genau die Datenstrukturen hat, die in der Klasse A vorgegeben sind. Das erweiterte Objekt a3 ist

3

35

dagegen allomorph zur Klasse A, vgl. Pfeil 62. Das Objekt a3 hat die allomorphe Klasse A.

Allomorphie ist die Fähigkeit der Objekte der Klasse A' so geführt zu werden, als wären sie Objekte ihrer allomorphen Klasse A, wenn diese Fähigkeit durch Maßnahmen auf der Seite des Anwendungsprogramms entsteht. Bei einer stufenweisen Erweiterung kann es mehr als eine allomorphe Klasse geben, z.B. die allomorphe Klasse der letzten Erweiterung und die allomorphe Klasse der vorletzten Erweiterung.

Ein erweitertes Objekt hat nur dann eine allomorphe Klasse, wenn das erweiterte Objekt ohne die Erweiterungen zur allomorphen Klasse kompatibel gemäß Standard X.720 Abschnitt

5.2.2 ist. Insbesondere hat das erweiterte Objekt demnach alle Attribute, Attributgruppen, Führungsfunktionen und Bestätigungsverfahren, die auch in der allomorphen Klasse festgelegt sind. Durch das Berücksichtigen der Allomorphie beim Erweitern der Klasse A wird erreicht, daß das Führungsnetz 10 auch nach der Erweiterung fehlerfrei arbeitet.

Figur 3 zeigt das Bearbeiten von Nachrichten im Steuerrechner 16 nach der Weiterentwicklung der Klasse A zur Klasse A', wobei das Objekt a2 geführt wird, das vor der Weiterentwicklung als zur Klasse A gehörend angelegt worden ist. Beim Weiterentwickeln der Klasse A zur Klasse A' wurde Allomorphie berücksichtigt, so daß Objekte der Klasse A' zur Klasse A allomorph sind. Außerdem wurden beim Weiterentwickeln alle Objekte der Klasse A' umgewandelt, indem Datenfelder und Methoden ergänzt worden sind.

Der Steuerrechner 16 enthält ein Schnittstellenprogramm 100, das vom Bedienrechner 24 kommende Wartungsnachrichten, beispielsweise eine Wartungsnachricht WN1, bearbeitet und das Bestätigungsnachrichten, beispielsweise eine Bestätigungsnachrichten, beispielsweise eine Bestätigungsnachricht BN1', an den Bedienrechner 24 sendet. Das Schnittstellenprogramm 100 ist die Schnittstelle zwischen dem Be-

dienrechner 24 und mehreren Anwendungsprogrammen im Steuerrechner 16, von denen in Figur 3 zwei Anwendungsprogramme 102
und 104 gezeigt sind. Das Anwendungsprogramm 102 dient zum
Verwalten der zu den an die Vermittlungsstelle 16 angeschlossenen Teilnehmern Tln2, Tln3 gehörenden Daten. Das Anwendungsprogramm 104 wird zur Verkehrsmessung verwendet.

Die vom Bedienrechner kommende Wartungsnachricht WN1 wird beim Abarbeiten des Schnittstellenprogramms 100 an das Anwendungsprogramm 102 als geänderte Wartungsnachricht WN1' weitergeleitet. Für das Anwendungsprogramm 104 bestimmte Wartungsnachrichten werden dagegen beim Abarbeiten des Schnittstellenprogramms 100 an das Anwendungsprogramm 104 weitergeleitet, vgl. Pfeil 106.

15

20

Nach dem Bearbeiten der Wartungsnachricht WN1' im Anwendungsprogramm 102 erzeugt das Anwendungsprogramm 102 eine Bestätigungsnachricht BN1 für die Schnittstelle 100. Hat das Anwendungsprogramm 104 eine Wartungsnachricht bearbeitet, so sendet es ebenfalls eine Bestätigungsnachricht an das Schnittstellenprogramm 100, vgl. Pfeil 108.

Beim Bearbeiten der Wartungsnachricht WN1' arbeitet das Anwendungsprogramm 102 mit einem ebenfalls im Steuerrechner 36 25 vorhandenen Datenbankprogramm 110 zusammen, mit dem Teilnehmerdaten im Speicher 42 gespeichert, verändert, gelöscht oder gelesen werden. Das Anwendungsprogramm 102 sendet Anforderungen in Form von Nachrichten an das Datenkbankprogramm 110, beispielsweise die Nachricht N1. Nach dem Ausführen der An-30 forderung in der Nachricht N1 sendet das Datenbankprogramm 110 eine Ergebnisnachricht EN1 an das Anwendungsprogramm 102 zurück. Das Anwendungsprogramm 104 arbeitet mit einem eigenen Datenbankprogramm 112 zusammen, indem es Anforderungen an das Datenbankprogramm 112 sendet, vgl. Pfeil 114, und indem es 35 vom Datenbankprogramm 112 Ergebnisnachrichten empfängt und weiterbearbeitet, vgl. Pfeil 116.

Das Datenbankprogramm 110 benutzt für die Objekte der ursprünglichen Klasse A und die Objekte der weiterentwickelten Klasse A' dasselbe Zugriffsverfahren. Dies ist möglich, weil Kombinationsklassen verwendet werden, in denen die Datenstrukturen und Methoden der ursprünglichen Klasse und der erweiterten Klasse zusammengeführt sind. Eine Kombinationsklasse KA ist die Kombination der Klasse A und der Klasse A'. Sämtliche Objekte im Speicher 42, welche die allomorphe Klasse A als ursprüngliche Klasse haben, werden mit der Weiterentwicklung um die zusätzlichen Datenfelder der erweiterten Klasse A erweitert. Die zusätzlichen Datenfelder werden mit vorgegebenen Werten belegt.

Die Wartungsnachricht WN1 enthält ein Klassenkennzeichen moC, das die Klasse A als die Klasse angibt, auf welche sich die Wartungsnachricht WN1 bezieht. Ein Objektkennzeichen moI gibt das Objekt a2 an, auf das sich die Wartungsnachricht WN1 bezieht. Die Wartungsnachricht WN1 wird vom Bedienrechner 24 gesendet, um die Teilnehmerdaten des Teilnehmers Tln2 zu erfahren. Im Bedienrechner 24 ist lediglich bekannt, daß diese Teilnehmerdaten in dem Objekt a2 enthalten sind, das im Steuerrechner 36 gespeichert ist. Die Wartungsnachricht WN1 enthält weitere nicht dargestellte Datenfelder. In einem dieser Datenfelder ist beispielsweise die auszuführende Leseoperation festgelegt.

Beim Bearbeiten der Wartungsnachricht WN1 im Schnittstellenprogramm 100 wird die im Klassenkennzeichen moC angegebene
Klasse A ermittelt. An Hand der so ermittelten Klasse A wird
beim Abarbeiten des Schnittstellenprogramms 100 mittels einer
Tabelle T die Klasse A' als Ersatzklasse ermittelt und in das
Klassenkennzeichen moC der Wartungsnachricht WN1 eingetragen.
Die Tabelle T ist in einem Speicher 122 des Steuerrechners 36
gespeichert.

35

20

25

30

Die Wartungsnachricht WN1' betrifft mit dem Klassenkennzeichen moC=A' die erweiterte Klasse A'. Der Wert des Objekt-

kennzeichens moI=a2 bleibt in der Wartungsnachricht WN1' unverändert. Ein Kennzeichen allo in der Wartungsnachricht WN1' gibt alle Klassen an, die zur Klasse A' allomorph sind, d.h. im Ausführungsbeispiel die Klasse A. Das Schnittstellenprogramm 100 entnimmt diese Klassen ebenfalls der Tabelle T. Die anderen Daten der Wartungsnachricht WN1 werden in die Wartungsnachricht WN1' übernommen. Beim Erzeugen der zur Wartungsnachricht WN1 gehörenden Wartungsnachricht WN1' führt das Schnittstellenprogramm 100 auch eine Protokollanpassung von einem Übertragungsprotokoll auf der Übertragungsstrecke 22 in ein Nachrichtenprotokoll durch, das innerhalb des Steuerrechners 36 verwendet wird.

Die beim Bearbeiten der Wartungsnachricht WN1' durch das Anwendungsprogramm 102 erzeugte Nachricht N1 enthält einen Befehl B1, der angibt, daß Daten gelesen werden sollen. Als Parameter des Befehls B1 enthält die Nachricht N1 die Klasse
A', zu der die zu lesenden Daten gehören, sowie das Objekt
a2, dessen Daten gelesen werden sollen. Das Anwendungsprogramm 102 bearbeitet ausschließlich Nachrichten, die sich auf
Objekte der Klasse A' beziehen. Bezüglich der Klasse A sind
im Anwendungsprogramm 102 keine weiteren Maßnahmen getroffen.

Das Datenbankprogramm 110 greift beim Bearbeiten der Nach-25 richt N1 auf den Speicher 42 zu, um die Daten des Teilnehmers Tln2 zu lesen, die im Objekt a2 gespeichert sind. Das Objekt a2 enthält außerdem ein Ursprungskennzeichen oC, in dem die Klasse A angegeben ist, die beim Erzeugen des Objekts a2 angegeben worden ist. Das Datenbankprogramm 110 liest die mit 30 dem Befehl B1 angeforderten Daten im Objekt a2 und trägt diese Daten in die Ergebnisnachricht EN1 ein. In der Ergebnisnachricht EN1 ist außerdem mittels eines Antwortkennzeichens AB1 vermerkt, daß die Ergebnisnachricht EN1 das Ergebnis enthält, das beim Bearbeiten des Befehls B1 erzeugt worden ist. Weiterhin ist in der Ergebnisnachricht EN1 die Klasse A' als die Klasse angegeben, welche von der Ergebnisnachricht EN1 betroffen ist. Das Ursprungskennzeichen oC=A wird ebenfalls

25

30

vom Datenbankprogramm 110 in der Ergebnisnachricht EN1 an das Anwendungsprogramm 102 übermittelt.

Beim Abarbeiten des Anwendungsprogramms 102 wird vorausge5 setzt, daß sich alle zu bearbeiteten Nachrichten auf die
Klasse A' und nicht auf die Klasse A beziehen. Durch das Anwendungsprogramm 102 wird der Wert des Ursprungskennzeichens
oC=A als Wert des Klassenkennzeichens moC in der Bestätigungsnachricht BN1 übernommen. Für diese Wertzuweisung ist es
10 nicht erforderlich, daß das Anwendungsprogramm 102 Objekte
der Klasse A bearbeiten kann. Außerdem enthält die Bestätigungsnachricht BN1 das Ursprungskennzeichen oC=A und die abgefragten Teilnehmerdaten des Teilnehmers Tln2. Das Schnittstellenprogramm 100 braucht sich aufgrund dieser Vorgehens15 weise das Klassenkennzeichen der Wartungsnachricht WN1 nicht
zu merken.

Die Bestätigungsnachricht BN1 wird vom Schnittstellenprogramm 100 bearbeitet und als Bestätigungsnachricht BN1' gemäß dem Übertragungsprotokoll auf der Übertragungsstrecke 22 an den Bedienrechner 24 gesendet. An Hand der Tabelle T ermittelt das Schnittstellenprogramm 100, welche Datenfelder in der Nachricht BN1 nicht in Objekten der Klasse A enthalten sind. Diese Datenfelder werden nicht in die Bestätigungsnachricht BN1' übernommen.

Der Bedienrechner 24 empfängt die Bestätigungsnachricht BN1' und kann sie wie eine Nachricht bearbeiten, die sich auf ein Objekt der Klasse A bezieht. Objekte der Klasse A' im Steuerrechner 36 werden vom Bedienrechner 24 aus so geführt, als wären sie Objekte der Klasse A. Die Weiterentwicklung im Steuerrechner 36 schränkt die Betriebsmerkmale des Bedienrechners 24 somit nicht ein.

Figur 4 zeigt das Bearbeiten von Nachrichten im Steuerrechner 16 nach der Weiterentwicklung der Klasse A zur Klasse A', wobei das Objekt a3 geführt wird, das erst nach der Weiterentwicklung erzeugt wird. Durch die Eingaben einer Bedienperson in den Bedienrechner 24 wird im Bedienrechner 24 eine Wartungsnachricht WN2 erzeugt, welche die Aufgabe hat, im Steuerechner 16 das Objekt a3 für die Teilnehmerdaten des Teilnehmers Tln3 zu erzeugen. Die Wartungsnachricht WN2 enthält deshalb in einem Befehlsfeld eine Codierung für den Befehl "Erzeuge". Das Klassenkennzeichen moC in der Wartungsnachricht WN2 kennzeichnet die Klasse A als die Klasse, auf die sich die Wartungsnachricht WN2 bezieht. Das Objektkennzeichen moI der Wartungsnachricht WN2 kennzeichnet das Objekt a3, das durch die Wartungsnachricht WN2 erzeugt werden soll. Die Wartungsnachricht WN2 enthält außerdem noch weitere nicht dargestellte Daten.

15 Das Schnittstellenprogramm 100 bearbeitet die Wartungsnachricht WN2 ebenso wie oben für die Wartungsnachricht WN1 erläutert. Beim Bearbeiten der Wartungsnachricht WN2 wird wiederum die im Speicher 122 gespeicherte Tabelle T verwendet, um für die im Klassenkennzeichen moC der Wartungsnachricht 20 WN2 angegebene Klasse A die Ersatzklasse A' zu ermitteln und als Klassenkennzeichen moC der Wartungsnachricht WN2' zu verwenden. Das Objektkennzeichen moI=a3 hat in der Wartungsnachricht WN2' den gleichen Wert, wie in der Wartungsnachricht WN2. Auch die verbleibenden Daten werden aus der Wartungs-25 nachricht WN2 beim Ausführen des Schnittstellenprogramms 100 in die Wartungsnachricht WN2' übernommen. Zusätzlich wird im Kennzeichen allo der Wartungsnachricht WN2' vermerkt, daß die Klasse A die Klasse ist, zu der die Klasse A' allomorph ist. Die so erzeugte Wartungsnachricht WN2' wird vom Schnittstellenprogramm 100 gemäß dem Protokoll im Steuerrechner 36 an 30 das Anwendungsprogramm 102 gesendet.

Das Anwendungsprogramm 102 sendet beim Bearbeiten der Wartungsnachricht WN2' eine Nachricht N2 an das Datenbankprogramm 110, um das Objekt a3 im Speicher 42 anlegen zu lassen. Die Nachricht N2 enthält verschlüsselt den Befehl "Erzeuge",

25

die erweiterte Klasse A' sowie den Namen des anzulegenden Objekts a3.

Beim Bearbeiten der Nachricht N2 mittels des Datenbankprogramms 110 wird im Speicher 42 das Objekt a3 erzeugt, indem
diesem Objekt bestimmte Speicherzellen zugewiesen werden, die
mit Anfangswerten belegt werden. Im Ursprungskennzeichen oC
des Objekts a3 wird die Klasse A' vermerkt, weil diese Klasse
beim Erzeugen des Objektes a3' angegeben wurde. Ein Kennzeichen allo im Objekt a3 weist auf die Klasse A hin, weil das
Objekt a3 zu der Klasse A allomorph ist.

Das Datenbankprogramm 110 erzeugt eine Ergebnisnachricht EN2, um das Erzeugen des Objekts a3 zu bestätigen. Die Ergebnisnachricht EN2 enthält ein Antwortkennzeichen AB2, das angibt, daß die Ergebnisnachricht EN2 beim Bearbeiten eines Befehls "Erzeuge" entstanden ist. Weiterhin enthält die Ergebnisnachricht EN2 ein Kennzeichen, das auf die Klasse A' hinweist, das Ursprungskennzeichen oC=A', das Kennzeichen allo=A sowie weitere nicht dargestellte Daten des Objekts a3.

Die Ergebnisnachricht EN2 wird vom Anwendungsprogramm 102 bearbeitet, wobei eine Bestätigungsnachricht BN2 erzeugt wird. In der Bestätigungsnachricht BN2 wird als Wert des Klassenzeichens moC der Wert des Ursprungskennzeichens oC=A' verwendet. Die anderen Daten der Ergebnisnachricht EN2 werden in die Bestätigungsnachricht BN2 übernommen.

Beim Abarbeiten des Schnittstellenprogramms 100 wird nach dem 30 Empfang der Bestätigungsnachricht BN2 eine Bestätigungsnachricht BN2' gemäß dem auf der Übertragungsstrecke 22 verwendeten Übertragungsprotokoll erzeugt. Die Bestätigungsnachricht BN2' enthält alle Daten der Bestätigungsnachricht BN2, weil das Schnittstellenprogramm 100 an Hand der Tabelle T feststellt, daß keine Datenfelder entfernt werden müssen, wenn die Bestätigungsnachricht BN2 als Klassenkennzeichen die Klasse A' hat.

Beim Bearbeiten der Bestätigungsnachricht BN2' im Bedienrechner 24 wird an Hand des Kennzeichens allo festgestellt, daß die Bestätigungsnachricht BN2' Daten eines Objekts der Klasse A betrifft bzw. daß die Bestätigungsnachricht BN2' so behandelt werden kann, als würde sie Daten eines Objektes der Klasse A enthalten. Obwohl das Programm im Bedienrechner 24 nach der Weiterentwicklung im Steuerrechner 36 nicht verändert worden ist, können Objekte der Klasse A' im Steuerrechner 36 vom Steuerrechner 24 aus so geführt werden, als wären sie Objekte der Klasse A. Solange im Bedienrechner A nur die Klasse A bekannt ist, werden in der Bestätigungsnachricht BN2' auch nur die in Objekten der Klasse A enthaltenen Daten bearbeitet.

15

Aufgrund des an Hand der Figuren 3 und 4 erläuterten Verfahrens ist es auch möglich, Klassen zu unterstützen, die zu mehreren Klassen allomorph sind. Wird z.B. die Klasse A' zu einer Klasse A' weiterentwickelt, so können Objekte zu den Klassen A' und A allomorph sein. Der Steuerrechner 36 kann dann von Bedienrechnern 24 ausgeführt werden, bei deren Programmierung mindestens eine der Klassen A, A' oder A'' bekannt war. Auf eine Unterstützung der Klasse A kann verzichtet werden, sobald alle Bedienrechner zumindest die Klasse A' kennen.

Das an Hand der Figuren 3 und 4 erläuterte Verfahren wird bei Wartungsnachrichten zum Erzeugen und bei Wartungsnachrichten zum Lesen von Objekten angewendet. Außerdem können mit diesem Verfahren Daten in Objekten verändert sowie Objekte gelöscht werden. Damit diese Verfahren fehlerfrei arbeiten, muß der Bedienrechner 24 fähig sein:

Die Bestätigungsnachrichten BN den zugehörigen Wartungs nachrichten WN zuzuordnen,

...

10

20

25

30

. 35

allomorph ist.

- das Kennzeichen allo in den Bestätigungsnachrichten BN zu lesen und auszuwerten,
- unbekannte Namensbindungen (vgl. Erläuterungen zu Fig. 5)

 zu übergehen und unbekannte Kennzeichenwerte und Parameter zu übergehen,
 - falls im Bedienrechner eine Datenbank vorhanden ist die Ursprungsklasse eines Objekts zu speichern,

- unbekannte wahlfreie Werte zu übergehen, und

- unbekannte Werte vom Datentyp "enumerated" zu übergehen.
- 15 Mit anderen Worten muß der Bedienrechner 24 so programmiert sein, daß von ihm aus Objekte in einen Steuerrechner mit größerem Wissen geführt werden können.

Das Schnittstellenprogramm 100 hat folgende Merkmale:

- Beim Erweitern einer Klasse ist es erforderlich, die erweiterte Klasse in die Tabelle T aufzunehmen, das bedeutet, daß für jede erweiterte Klasse die Klassen gespeichert werden müssen, zu denen die erweiterte Klasse

- Die Namensbindungen, welche von der erweiterten Klasse betroffen werden, müssen in der Tabelle T gespeichert werden.
- Das Schnittstellenprogramm 100 soll aus den vom Anwendungsprogramm 102 kommenden Bestätigungsnachrichten BN die Daten entfernen, die nicht zur kompatiblen Klasse gehören, falls die an den Bedienrechner 24 gesendete Bestätigungsnachricht BN die nicht erweiterte Klasse betrifft.

20

25

30

- Das Schnittstellenprogramm 100 muß Filterbefehle bearbeiten können. Dies wird unten an Hand der Figur 5 noch genauer erläutert.
- Das Anwendungsprogramm 102 erfüllt die folgenden Anforderungen:
- Sobald zur erweiterten Klasse übergegangen wird, werden unabhängig vom Wissen des Bedienrechners nur noch Objekte der neuen Klasse erzeugt.
 - Die neuen Klassen sind entweder erweiterte Klassen oder Klassen, die mit den bisherigen Klassen nichts zu tun haben.

Das Datenbankprogramm 110 ist so ausgelegt, daß nach einer Weiterentwicklung der gesamte Datenbestand zur Ursprungsklasse in einen Datenbestand der erweiterten Klasse umgewandelt wird.

Das Erstellen der Tabelle T und das Umwandeln des Datenbestandes in der Datenbank des Datenbankprogramms 110 werden durch Hilfsprogramme unterstützt. Diese Hilfsprogramme werten Beschreibungssprachen aus, mit deren Hilfe die Erweiterung von Klassen angegeben wird.

Im folgenden werden Bedingungen angesprochen, die bei der Berücksichtigung von Allomorphie eingehalten werden müssen. Diese Bedingungen gelten auf der Ebene der Klassen, obwohl gemäß Standard X.720 Allomorphie zunächst eine Eigenschaft eines Objekts ist. Im Standard X.720 sind folgende Bedingungen angesprochen:

- Bedingungen für die erweiterte Klasse in Abschnitt 5.2.2.1,

5

15

- Bedingungen für sogenannte Programmpakete in den Abschnitten 5.2.2.1 und 5.2.2.2,
- Bedingungen für Kennzeichenwerte in den Abschnitten 5.2.2.3 und 5.3.4.1,
 - Bedingungen für sogenannte Kennzeichengruppen in Abschnitt 5.2.2.4,
- 10 -- Bedingungen für Aktionen, Bestätigungen und Parameter in Abschnitt 5.2.2.54,
 - Bedingungen für das Verhalten der Objekte in Abschnitt 5.2.2.6, und
 - Bedingungen für die Namensbindung in Abschnitt 5.3.4.1.

Figur 5 zeigt in einem Teil a einen Ausschnitt aus einem sogenannten binären Baum, der die Namensbindung von Objekten der Klasse A und A' festlegt. Namensbindung ist die für das Festlegen eines eindeutigen Namens für ein Objekt verwendete Zuordnung des Objektes zu einem sogenannten übergeordneten Objekt. Gleiche Objektnamen für unterschiedliche Objekte können verwendet werden, wenn die Objekte jeweils zu einem ande-25 ren übergeordneten Objekt gehören. Aus dem Namen des übergeordneten Objekts und dem Namen der so untergeordneten Objekte entsteht dann ein im Steuerrechner 16 eindeutiger Name. Das übergeordnete Objekt ist beim Erzeugen eines Objektes anzugeben. Das untergeordnete Objekt wird als im übergeordneten Ob-30 jekt enthalten angesehen, englisch als "Containment" bezeichnet.

Ein Objekt bl der Klasse B ist das übergeordnete Objekt für die Objekte a2 und a3, die beide nach der Weiterentwicklung zur Klasse A' gehören, jedoch unterschiedliche ursprüngliche Klassen oC=A bzw. oC=A' haben. In der Tabelle T, die vom Schnittstellenprogramm 100 verwendet wird, sind die Namens-

bindungen vermerkt. Für den im Teil A der Figur 5 gezeigten Ausschnitt des Namensbaumes ist in der Tabelle T vermerkt, daß Objekte der Klasse B untergeordnete Objekte der ursprünglichen Klasse A bzw. der Klasse A' enthalten. Bei jeder Weiterentwicklung werden auch die Namensbindungen in der Tabelle T an den Namensbaum angepaßt, der nach der Weiterentwicklung maßgeblich ist.

Im Teil b der Figur 5 ist ein Zugriff auf Objekte beider

Klassen A und A' argestellt, nachdem auch der Bedienrechner

24 beide Klassen A und A' kennt. Eine Wartungsnachricht WN3

enthält eine Filteranweisung Fl=((oC=A)ODER(oC=A')), in der

festgelegt ist, daß untergeordnete Objekte der Klasse A oder

A' erfaßt werden sollen. Das Klassenkennzeichen moC=B der

Wartungsnachricht WN3 gibt an, daß sich die Wartungsnachricht

WN3 auf die Klasse B bezieht. Das Objektkennzeichen moI=bl

der Wartungsnachricht WN3 gibt an, daß das Objekt bl bearbei
tet werden soll, d.h. das übergeordnete Objekt ist.

- Beim Ausführen des Schnittstellenprogramms 100 in der Vermittlungsstelle 16 wird die Filteranweisung F1 unverändert an das Anwendungsprogramm 102 gesendet. Das bedeutet insbesondere, daß in der Filteranleitung F1 Operationen, die sich auf die ursprüngliche Klasse A beziehen, nicht durch Operationen ersetzt werden, die sich auf die erweiterte Klasse A' beziehen. Diese Maßnahme gewährleistet, daß der Bedienrechner 24 zwischen Objekten der ursprünglichen Klasse A und der ursprünglichen Klasse A' unterscheiden kann.
- Das Anwendungsprogramm 102 veranlaßt, daß mittels Datenbankprogramm 110 aus dem Speicher 42 die Daten der Objekte a2 und
 a3 gelesen werden, die als ursprüngliche Klasse die Klasse A
 bzw. A' haben. An das Schnittstellenprogramm 100 werden zwei
 nicht dargestellte Bestätigungsnachrichten BN3 und BN4 gesendet. Im Schnittstellenprogramm 100 wird aus der Bestätigungsnachricht BN3 eine neue Bestätigungsnachricht BN3' erzeugt,
 die das Klassenzeichen moC=A, das Objektkennzeichen moI=a2,

das Ursprungskennzeichen oC=A und das Hilfskennzeichen allo ={} enthält. Aus der Bestätigungsnachricht BN4 wird im Schnittstellenprogramm 100 eine Bestätigungsnachricht BN4' erzeugt, welche das Klassenkennzeichen moC=A', das Objekt-kennzeichen moI=a3, das Ursprungskennzeichen oC=A', und das Hilfskennzeichen allo={A} enthält.

Der CCITT-Standard X.734 "Information Technology - Open Systems Interconnection - Systems Management: Event Report Management Function" aus dem Jahre 1993 erläutert die Ereignissteuerung im Führungsnetz 10, vgl. Figur 1. Es werden sogenannte Diskriminatoren verwendet, die Ereignisse innerhalb der Vermittlungsstelle 16 nur unter bestimmten Bedingungen an den Bedienrechner 24 weiterleiten. Nach der Weiterentwicklung der Klasse A zur Klasse A' genügt es, die Diskriminatoren, welche die Klasse A betreffen, in Diskriminatoren umzuwandeln, welche die Klasse A' betreffen. Werden nach der Weiterentwicklung neue Diskriminatoren erzeugt, so wird die Klasse A durch die Klasse A' ersetzt, wenn die Klasse A als Auswahl-kriterium für das Weiterleiten der Nachrichten angegeben ist.

Patentansprüche

- 1. Verfahren zum Betreiben eines Telekommunikationsnetzes (10, 12),
- 5 bei dem ein Netzelement (16) an einem Netzknoten eines Telekommunikationsnetzes (12) von einem Steuerrechner (36) gesteuert wird,
 - im Steuerrechner (36) neben dem Betriebssystem mehrere Anwendungsprogramme (102, 104) gespeichert sind, bei deren Aus-
- führen Anwendungsobjekte (a2, a3) bearbeitet werden, die Anwendungsobjekte (a2, a3) je nach Zugehörigkeit zu einer Klasse (A, A') Daten mit einer vorgegebenen Datenstruktur sowie vorzugsweise auch vorgegebene Verfahren zum Bearbeiten der Daten haben,
- zwischen einem Bedienrechner (24) und dem Steuerrechner (36) eine Verbindung aufgebaut wird, über die der Steuerrechner (36) mittels mindestens einer Wartungsnachricht (WN1 bis WN3) gewartet wird,
- die Wartungsnachricht (WN1 bis WN3) ein Klassenkennzeichen (moC) enthält, das die Wartungsnachricht (WN1 bis WN3) einer Klasse (A, A') zuordnet,
 - das Klassenkennzeichen (moC) der Wartungsnachricht (WN1 bis WN3) die im Bedienrechner bekannte Klasse (A) eines zu bearbeitenden Anwendungsobjektes (a2, a3) angibt,
- beim Abarbeiten des Schnittstellenprogramms (100) anhand des Klassenkennzeichens (moC) ein Ersatzkennzeichen ermittelt wird, welches eine Ersatzklasse (A') angibt, der das zu bearbeitende Anwendungsobjekt (a2, a3) im Netzelement (16) angehört,
- 30 beim Abarbeiten des Schnittstellenprogramms (100) das Ersatzkennzeichen (A') in eine geänderte Wartungsnachricht (WN1' bis WN2') aufgenommen wird,
 - und bei dem beim Bearbeiten der geänderten Wartungsnachricht (WN1' bis WN2') durch ein Anwendungsprogramm (102) das zu be-
- arbeitende Anwendungsobjekt (a2, a3) als Objekt der Ersatzklasse (A') bearbeitet wird.

- 2. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß beim Ermitteln des Ersatzkennzeichens (A') eine im Speicher (122) des Steuerrechners
 (16) gespeicherte erste Tabelle (T) verwendet wird, in der
 dem Klassenkennzeichen (A) ein Ersatzkennzeichen (A') zugeordnet ist.
- 3. Verfahren nach Anspruch 1 oder 2, dadurch gekennz'eichnet, daß das Anwendungsprogramm (102) nach dem Bearbeiten der geänderten Wartungsnachricht (WN1') eine Bestätigungsnachricht (BN1) erzeugt, in der die beim Erzeugen des zu
 bearbeitenden Anwendungsobjekts (a2, a3) angegebene Klasse
 (oC) als Klassenkennzeichen (moC) angegeben ist.
- 4. Verfahren nach Anspruch 3, dadurch gekennzeichnet, daß beim Abarbeiten des Schnittstellenprogramms (100) aus der Bestätigungsnachricht (BN1) eine geänderte Bestätigungsnachricht (BN1') erzeugt wird, die nur solche Daten enthält, die ein Anwendungsobjekt (a2) der Klasse (A) hat, auf die sich die Bestätigungsnachricht (BN1) bezieht, und daß die geänderte Bestätigungsnachricht (BN1') an den Bedienrechner (24) gesendet wird.
- 5. Verfahren nach Anspruch 3 oder 4, dadurch gekenn25 zeichnet, daß die beim Erzeugen des zu bearbeitenden Anwendungsobjektes (a2, a3) angegebene Klasse (A, A') als Ursprungsklasse (oC) in den Daten das zu bearbeitenden Anwendungsobjekts (a2, a3) gespeichert ist,
 und daß beim Abarbeiten des Anwendungsprogramms (102) die Ur30 sprungsklasse (oC) als Klassenkennzeichen (moC) verwendet
 wird.
- 6. Verfahren nach einem der Ansprüche 3 bis 5, dadurch gekennzeichnet, daß die Bestätigungsnachricht (BN1)

 ein Hilfskennzeichen (allo) enthält, in welchem mindestens eine Klasse (A) bezeichnet ist, die im Bedienrechner (24) und/oder in mindestens einem anderen Bedienrechner als die

Klasse (A) bekannt ist, zu der das zu bearbeitende Anwendungsobjekt (a2, a3) gehört, vorzugsweise zumindest die im Bedienrechner (24) bekannte Klasse (A) des zu bearbeitenden Anwendungsobjekts (a2, a3).

5

7. Verfahren nach Anspruch 6, dadurch gekennzeichnet, daß die Bestätigungsnachricht (BN1) neben dem Klassenkennzeichen (moC) ein Ursprungskennzeichen (oC) enthält, in welchem die Ursprungsklasse (A, A') angegeben ist.

10

- 8. Verfahren nach Anspruch 6 oder 7, dadurch gekennzeichnet, daß mindestens eine im Bedienrechner (24)
 und/oder in mindestens einem anderen Bedienrechner für das
 Anwendungsobjekt bekannte Klasse (A) als Allomorphklasse (allo) in den Daten des Anwendungsobjektes (a2, a3) gespeichert
 ist,
- und daß beim Abarbeiten des Anwendungsprogramms (102) die Allomorphklasse (allo) als Hilfskennzeichen (allo) verwendet wird.

20

25

15

- 9. Verfahren nach einem der Ansprüche 4 bis 8, dadurch gekennzeichnet, daß die bzw. eine beim Abarbeiten des Schnittstellenprogramms (100) für den Bedienrechner (24) erzeugte Bestätigungsnachricht (BN1') das Hilfskennzeichen (allo) und/oder das Klassenkennzeichen (moC) und/oder das Ursprungskennzeichen (oC) enthält.
- 10. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß das Netzelement eine Ver30 mittlungsstelle (16), ein Cross-Connector oder eine Konzentratoreinheit ist.
- 11. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß das Telekommunikationsnetz
 35 (12) ein Festnetz, ein Mobilfunknetz oder ein Netz mit einem
 Festnetzanteil und einem Mobilfunknetzanteil ist.

12. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß das Schnittstellenprogramm
(100) weitere Schnittstellenfunktionen zwischen dem Bedienrechner (24) und den Anwendungsprogrammen (102, 104) ausführt,

vorzugsweise eine Ereignissteuerung zum Festlegen der Bearbeitungsreihenfolge der Wartungsnachrichten (WN1 bis WN3) und/oder eine Anpassung der vom Bedienrechner (24) kommenden Nächrichten (WN1 bis WN3) an ein Protokoll zur Nachrichtenübertragung innerhalb des Steuerrechners (16),

und/oder eine Anpassung der von den Anwendungsprogrammen (102, 104) kommenden Bestätigungsnachrichten (BN1, BN2) an ein vorgegebenes Protokoll zur Nachrichtenübertragung zwischen Bedienrechner (24) und Steuerrechner (16).

15

5

:

٤.

13. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß ein erstes Anwendungsprogramm (102) zur Teilnehmerverwaltung,

und/oder ein zweites Anwendungsprogramm zum Verwalten von Verbindungsleitungen zu anderen Vermittlungseinrichtungen, und/oder ein drittes Anwendungsprogramm zur Wartung der Vermittlungseinrichtungen,

und/oder ein viertes Anwendungsprogramm (104) zur Verkehrsmessung der geschalteten Verbindungen (18) verwendet wird.

25

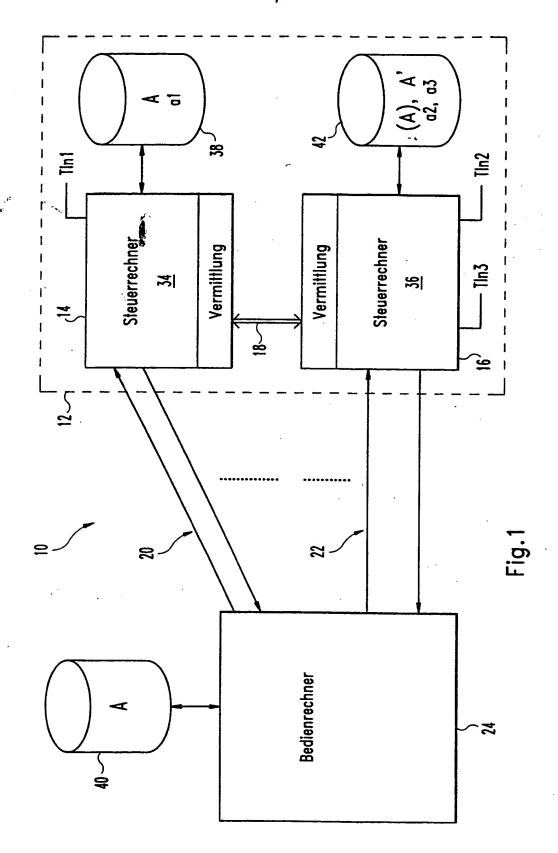
35

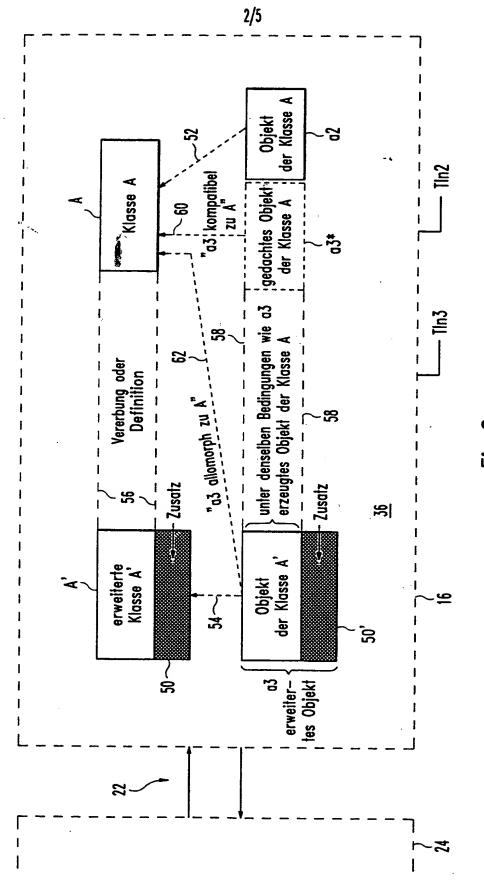
- 14. Verfahren nach Anspruch 13, dadurch gekennzeichnet, daß die Anwendungsobjekte (a2, a3) des ersten Anwendungsprogramms (102) für jeweils einen Teilnehmer (Tln2,
 Tln3) die Teilnehmerdaten enthalten, vorzugsweise die Rufnummer und/oder die nutzbaren Telekommunikationsdienste.
- 15. Verfahren nach einem der vorhergehenden Ansprüche, da-durch gekennzeichnet, daß die Wartungsnachrichten (WN1 bis WN3) ein Namenskennzeichen (moI) für den Namen des Anwendungsobjektes (a2, a3) enthalten, auf welches sich die Wartungsnachricht (WN1 bis WN3) bezieht.

- 16. Netzelement (16) zum Betreiben eines Telekommunikationsnetzes (10, 12), dadurch gekennzeichnet, daß es einen Speicher zum Speichern von Befehlen hat, bei deren Abarbeiten das Verfahren nach einem der vorhergehenden Ansprüche
 durchgeführt wird.
- 17. Telekommunikationsnetz (10, 12), gekennzeichnet durch, ein Netzelement gemäß Anspruch 16.

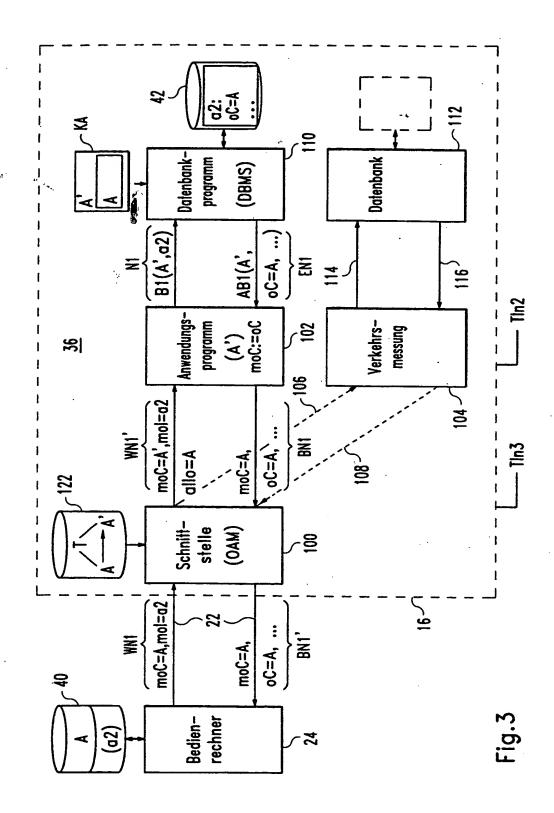
- 18 cm

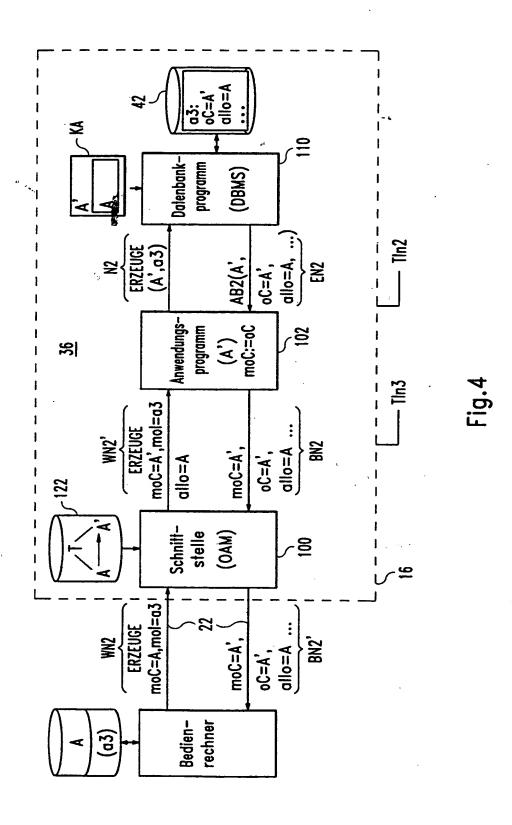
... ...





. ig. 2





16.

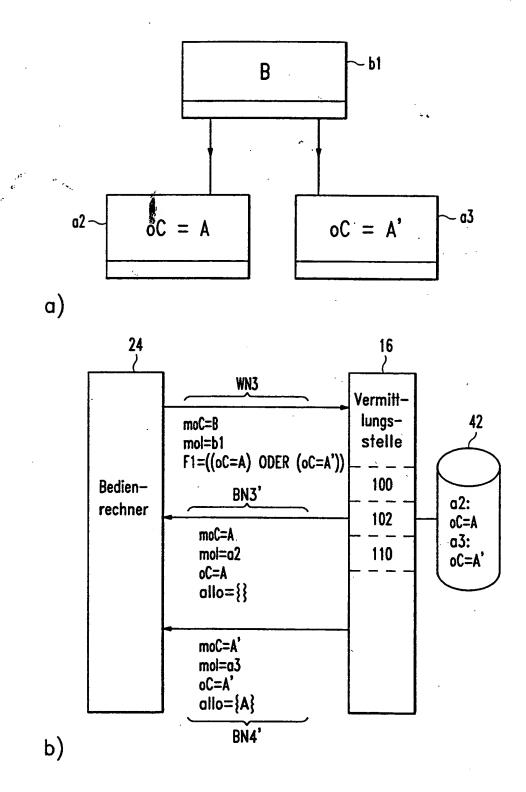


Fig.5

INTERNATIONAL SEARCH REPORT

		PCT/L. 00/	,	
A CLASSII IPC 7	FICATION OF SUBJECT MATTER H04Q3/00			
-				
According to	International Patent Classification (IPC) or to both national classification and IPC			
	SEARCHED cumentation searched (classification system followed by classification symbols)			
IPC 7	H04Q			
Documented	ion searched other than minimum documentation to the extent that such documents a	re included in the fields se	arched	
Electronic d	ata base consulted during the international search (name of data base and, where pr	actical, search terms used)	
j:				
e) No	*			
C. DOCUMI	ENT'S CONSIDERED TO BE RELEVANT			
Category *	Citation of document, with indication, where appropriate, of the relevant passages		Relevant to claim No.	
X	EP 0 817 422 A (TELEFONAKTIEBOLAGET LM ERICSSON) 7 January 1998 (1998-01-07)		1,10,11, 16,17	
A	abstract column 1, line 22 -column 2, line 50 column 4, line 55 -column 5, line 18 column 7, line 31 -column 8, line 18	2 -9 , 12-15		
	·			
X Furt	ther documents are listed in the continuation of box C. X Petent	family members are listed	in annex.	
*Special categories of cited documents: "T" later document published after the international filing date or priority date and not in conflict with the application but				
Considered to be of particular relevance citied to understand the principle or theory underlying the invention considered to be of particular relevance citied to understand the principle or theory underlying the invention				
filing date Cannot be considered novel or cannot be considered to involve an inventor step when the document is taken alone L document which may throw doubts on priority claim(s) or involve an inventor step when the document is taken alone				
which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or document is combined with one or more other such document."				
colument resuming to an oral disclosure, use, exhibition of colument is combined with one or more other such document other means. "P" document published prior to the international filing date but in the art. Later than the priority date claimed "&" document member of the earne patent family				
Date of the		ling of the International sec		
		05/2000		
Mama and	mailing actions of the ICA			

Larcinese, C

INTERNATIONAL SEARCH REPORT

eri. nal Application No

HERMANN A ET AL: "PHAMOS - PHILIPS ADVANCED MANAGEMENT AND OPERATIONS SYSTEM - FUNCTIONALITY AND ARCHITECTURE" PHILIPS TELECOMMUNICATION REVIEW, vol. 51, no. 1, 1 March 1993 (1993-03-01), pages 30-42, XPO00426372 page 32, left-hand column, line 16 -right-hand column, line 9 - line 43 page 38, left-hand column, line 9 - line 43 page 38, left-hand column, line 1 -page 39, left-hand column, line 4 -right-hand column, line 8 BERGHOLM J O ET AL: "SERVICE DESIGN AND INVENTORY SYSTEM-AN OBJECT-ORIENTED REUSABLE SOFTWARE ASSET" AT & T TECHNICAL JOURNAL, vol. 75, no. 2, 1 April 1996 (1996-04-01), pages 47-57, XPO00584939 ISSN: 8756-2324 page 49, left-hand column, line 12 -page 50, right-hand column, line 19 -page 51, right-hand column, line 19 -page 53, right-hand column, line 19 -page 54, right-hand column, line 8	ADVANCED MANAGEMENT AND OPERATIONS SYSTEM - FUNCTIONALITY AND ARCHITECTURE" PHILIPS TELECOMMUNICATION REVIEW, vol. 51, no. 1, 1 March 1993 (1993-03-01), pages 30-42, XP000426372 page 32, left-hand column, line 16 -right-hand column, line 9 - line 43 page 38, left-hand column, line 1 -page 39, left-hand column, line 2 page 40, left-hand column, line 4 -right-hand column, line 8 BERGHOLM J O ET AL: "SERVICE DESIGN AND INVENTORY SYSTEM-AN OBJECT-ORIENTED REUSABLE SOFTWARE ASSET" AT & T TECHNICAL JOURNAL, vol. 75, no. 2, 1 April 1996 (1996-04-01), pages 47-57, XP000584939 ISSN: 8756-2324 page 49, left-hand column, line 12 -page 50, right-hand column, line 12 page 53, right-hand column, line 19 -page	ory * Citation of document, with indication, where appropriate, of the re	plevant passages	Relevant to claim No.
page 40, left-hand column, line 4 -right-hand column, line 8 BERGHOLM J O ET AL: "SERVICE DESIGN AND INVENTORY SYSTEM-AN OBJECT-ORIENTED REUSABLE SOFTWARE ASSET" AT & T TECHNICAL JOURNAL, vol. 75, no. 2, 1 April 1996 (1996-04-01), pages 47-57, XP000584939 ISSN: 8756-2324 page 49, left-hand column, line 12 -page 50, right-hand column, line 12 page 53, right-hand column, line 19 -page	page 40, left-hand column, line 4 -right-hand column, line 8 BERGHOLM J O ET AL: "SERVICE DESIGN AND INVENTORY SYSTEM-AN OBJECT-ORIENTED REUSABLE SOFTWARE ASSET" AT & T TECHNICAL JOURNAL, vol. 75, no. 2, 1 April 1996 (1996-04-01), pages 47-57, XP000584939 ISSN: 8756-2324 page 49, left-hand column, line 12 -page 50, right-hand column, line 12 page 53, right-hand column, line 19 -page	ADVANCED MANAGEMENT AND OPERATION - FUNCTIONALITY AND ARCHITECTURING PHILIPS TELECOMMUNICATION REVIEW vol. 51, no. 1, 1 March 1993 (1902) pages 30-42, XP000426372 page 32, left-hand column, line right-hand column, line 90 page 35, left-hand column, line 43 page 38, left-hand column, line	ONS SYSTEM E" W, 993-03-01), 16 9 - line	
		page 40, left-hand column, line -right-hand column, line 8 BERGHOLM J O ET AL: "SERVICE DI INVENTORY SYSTEM-AN OBJECT-ORIE REUSABLE SOFTWARE ASSET" AT & T TECHNICAL JOURNAL, vol. 75, no. 2, 1 April 1996 (19 pages 47-57, XP000584939 ISSN: 8756-2324 page 49, left-hand column, line 50, right-hand column, line 12 page 53, right-hand column, line	ESIGN AND NTED 996-04-01), 12 -page e 19 -page	1-17

INTERNATIONAL SEARCH REPORT

informi

in patent family members

Inter: 1 Application No
PCT/ LP 00/02070

Detect document		D. Atlantin			
Patent document cited in search report	1	Publication date		Patent family member(s)	Publication date
				1.10.1201(0)	Cara
EP 817422	A	07-01-1998	AU	684254 B	11-12-1997
			AU	4988693 A	29-03-1994
			AU	670325 B	11-07-1996
			AU	4988893 A	29-03-1994
			AU	692695 B	11-06-1998
			AU	5248996 A	05-09-1996
			BR	9305624 A	01-03-1995
		•	BR	9306983 A	12-01-1999
		•	CA	2122334 A	17-03-1994
			, CA	2142015 A	17-03-1994
			CN	1088722 A	29-06-1994
			CN	1088375 A	22-06-1994
			DE	69322857 D	11-02-1999
	*		DE	69322857 T	27-05-1999
	4		EP	0788719 A	13-08-1997
	**		EP	0627143 A	07-12-1994
			ES	2126656 T	01-04-1999
•			FI	950890 A	27-02-1995
	4		FI	941944 A	2 7-04-1994
	7		GR	3029728 T	30 - 06-1999
			JP	8503342 T	09-04-1996
			JP	7503117 T	30-03-1995
			MX	9305186 A	31-01-1995
	•		NO	941406 A	18-04-1994
			NO	950710 A	26-04-1995
			SE	470456 B	11-04-1994
			SE	9300363 A	01-03-1994
•			WO	9406252 A	17-03-1994
			WO	9406232 A	17-03-1994
			SE.	9301000 A	01-03-1994
			SE	9303075 A	01-03-1994
			US	5634126 A	27-05-1997

INTERNATIONALE RECHERCHENBERICHT

ee Aktenzoichen PCT, __/ 00/02070

A. KLASSIFIZIERUNG DES ANNELDUNGSGEGENSTANDES IPK 7 H04Q3/00

Nach der Internationalen Patentiklassifikation (IPK) oder nach der nationalen Klassifikation und der IPK

B. RECHERCHIERTE GERIETE

Recherchierter Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole) IPK 7 H04Q

Recherchierte aber nicht zum Mindestprüfstoff gehörende Veröffentlichungen, soweit diese unter die recherchierten Gebiete fallen

Während der Internationalen Recherche konsultierte elektronische Datenbank (Name der Datenbank und evtl. verwendste Suchbegriffe)

C. ALS WESENTLICH ANGESEHENE UNTERLAGEN			
Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anapruch Nr.	
X	EP 0 817 422 A (TELEFONAKTIEBOLAGET LM ERICSSON) 7. Januar 1998 (1998-01-07) Zusammenfassung Spalte 1, Zeile 22 -Spalte 2, Zeile 50	1,10,11, 16,17	
A	Spalte 4, Zeile 55 -Spalte 5, Zeile 18 Spalte 7, Zeile 31 -Spalte 8, Zeile 18	2-9, 12-15	
-	-/		
;			
	,		

X	Welters Veröffentlichungen sind der Fortsetzung von Fel entnehmen	d C zu
Bee	ondere Kategorien von angegebenen Veröffentlichungen	

Siehe Anhang Patentiamilie

- "A" Veröffentlichung, die den allgemeinen Stand der Technik definiert, aber nicht als beeonders bedeutsam anzurehen ist
- "E" älteree Dokument, das jedoch enst am oder nach dem internationalen Anmeldedatum veröffentlicht worden ist
- "Veröffentlichung, die geeignet ist, einen Prioritätsanspruch zweitelhaft erscheinen zu lassen, oder durch die das Veröffentlichungsdatum einer anderen im Recherchenbericht genannten Veröffentlichung belegt werden. soll oder die aus einem anderen besonderen Grund angegeben ist (wie
- *O" Veröffentlichung, die sich auf eine m\u00e4ndliche Offenbarung, eine Benutzung, eine Ausstellung oder andere Ma\u00ednahmen bezieht "P" Ver\u00f6fentlichung, die vor dem internationalen 'Anmeldedatum, aber nach dem beanspruchten Priorit\u00e4ndatum ver\u00f6ffentlicht worden ist
- T Spätere Veröffentlichung, die nech dem internationalen Anmeldedatum oder dem Prioritätsdatum veröffentlicht worden ist und mit der Anmeldung nicht kollidiert, sondem nur zum Veretändnie des der Erfindung zugrundellegenden Prinzips oder der ihr zugrundellegenden Theorie angegeben ist
- "X" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann allein aufgrund dieser Veröffentlichung nicht als neu oder auf erfindertacher i ätigkeit beruhend betrachtst werden
- Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann nicht eis auf erfinderischer Tätigkeit beruhend betrachtet werden, wenn die Veröffentlichung mit ehner oder mehrenen anderen Veröffentlichungen dieser Kategorie in Veröffentlichung gebracht wird und diese Verbindung für einen Fachmann nahellegend ist
- *&* Veröffentlichung, die Mitglied derselben Patentfamilie ist

Absendedatum des Internationalen Recherchenberichts

Datum des Abschlusses der Internationalen Recherchs

9. Mai 2000

19/05/2000

Name und Postanschrift der Internationalen Recherchenbehörde

Fac: (+31-70) 340-3016

Europäischee Patentams, P.B. 5818 Patentiaan 2 NL – 2280 HV Fijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl.

Bevolimächtigter Bediensteter

Larcinese, C

Formblatt PCT/ISA/210 (Blatt 2) (Juli 1992)

INTERNATIONALE RECHERCHENBERICHT

T/EP 00/02070

	ng) ALS WESENTLICH ANGESEHENE UNTERLAGEN		
etegorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht komm	enden Teile	Betr. Anapruch Nr.
Å.	HERMANN A ET AL: "PHAMOS - PHILIPS ADVANCED MANAGEMENT AND OPERATIONS SYSTEM - FUNCTIONALITY AND ARCHITECTURE" PHILIPS TELECOMMUNICATION REVIEW, Bd. 51, Nr. 1, 1. März 1993 (1993-03-01), Seiten 30-42, XP000426372 Seite 32, linke Spalte, Zeile 16 -rechte Spalte, Zeile 9 Seite 35, linke Spalte, Zeile 9 - Zeile 43 Seite 38, linke Spalte, Zeile 1 -Seite 39, linke Spalte, Zeile 2 Seite 40, linke Spalte, Zeile 4 -rechte Spalte, Zeile 8		1-17
A	BERGHOLM J O ETSAL: "SERVICE DESIGN AND INVENTORY SYSTEM—AN OBJECT—ORIENTED REUSABLE SOFTWARE ASSET" AT & T TECHNICAL JOURNAL, Bd. 75, Nr. 2, 1. April 1996 (1996—04—01), Seiten 47-57, XP000584939 ISSN: 8756—2324 Seite 49, linke Spalte, Zeile 12 —Seite 50, rechte Spalte, Zeile 12 Seite 53, rechte Spalte, Zeile 19 —Seite 54, rechte Spalte, Zeile 8		1–17
			,
		·	
		•	
	·		
`			
·			
1			

INTERNATIONALE RECHERCHENBERICHT

Angaben zu Veröffentlichungen, die zu

Patentiamilie gehören

PCT/E: 00/02070

Im Recherchenbericht Datum der angeführtes Patentickument Veröffentlichung		Mitglied(er) der Patentfamilie	Datum der Veröffentlichung
EP 817422 A	07-01-1998	AU 684254 B	11-12-1997
1		AU 4988693 A	2 9-0 3-19 94
		AU 670325 B	11-07-1996
1		AU 4988893 A	29-03-1994
		AU 692695 B	11-06-1998
,		AU 5248996 A	0 5-09- 1996
		BR 9305624 A	01 -03-1995
		BR 9306983 A	12-01-1999
		CA 2122334 A	` 17-03-1994
		CA 2142015 A	17-03-1994
		CN 1088722 A	29-06-1994
		CN 1088375 A	22-06-1994
6 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		DE 69322857 D	11-02-1999
\$	₹ `	DE 69322857 T	2 7- 05 - 1999
·		EP 0788719 A	13-08-1997
	*	EP 0627143 A	07-12-1994
	•	ES 2126656 T	01-04-1999
		FI 950890 A	27-02-19 95
i .		FI 941944 A	27-04-1994
	>	GR 3029728 T	30-06-1999
		JP 8503342 T	09-04-1996
		JP 7503117 T	3 0- 03 -1995
·		MX 9305186 A	31 - 01 -1995
		NO 941406 A	18-04-19 94
		NO 950710 A	26 - 04-1995
1		SE 470456 B	11-04-1994
}		SE 9300363 A	01-03-1994
		WO 9406252 A	17-03-19 94
		WO 9406232 A	17 – 03–1994
]		SE 9301000 A	01-03-1994
		SE 9303075 A	01-03-19 94
17		US 5634126 A	2 7- 05 - 19 97

VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AU DEM GEBIET DES PATENTWES

PCT

INTERNATIONALER RECHERCHENBERICHT

(Artikel 18 sowle Regeln 43 und 44 PCT)

Aktenzeichen des Anmelders oder Anwalts	WEITERES siehe Mittellung über die Übermittlung des Internationalen Recherchenberichts (Formblatt PCT/ISA/220) sowie, soweit			
99P1391P	VORGEHEN zutreffend, na	chstehender Punkt 5		
Internationales Aktenzeichen	Internationales Anmeldedatum (Tag/Monat/Jahr)	(Frühestes) Prioritätsdatum (Tag/Monat/Jahr)		
PCT/EP 00/02070	09/03/2000	10/03/1999		
Anmelder				
SIEMENS AKTIENGESELLSCHAFT				
Dieser Internationale Recherchenbericht wurde von der Internationalen Recherchenbehörde erstellt und wird dem Anmelder gemäß Artikel 18 übermitteit. Eine Kopie wird dem Internationalen Büro übermitteit. Dieser Internationale Recherchenbericht umfaßt insgesamt				
1. Grundlage des Berichts		·		
a. Hinsichtlich der Sprache ist die inter	nationale Recherche auf der Grundlag ereicht wurde, sofern unter diesem Pur	e der Internationalen Anmeldung in der Sprache ikt nichts anderes angegeben ist.		
Die internationale Recherch Anmeldung (Regel 23.1 b)) o	e ist auf der Grundlage einer bei der Be durchgeführt worden.	phörde eingereichten Übersetzung der Internationalen		
Recherche auf der Grundlage des S	n Anmeldung offenbarten Nucleotid- u equenzprotokolis durchgeführt worden, dung in Schriflicher Form enthalten ist.	ınd/oder Aminosāuresequenz ist die Internationale , das		
	nalen Anmeldung in computerlesbarer	Form eingereicht worden ist.		
bei der Behörde nachträglich	n in schriftlicher Form eingereicht worde	en lst.		
bei der Behörde nachträglich	n in computertesbarer Form eingereicht	worden ist.		
Die Erklärung, daß das nach Internationalen Anmeldung is	träglich eingereichte schriftliche Seque m Anmeldezeitpunkt hinausgeht, wurde	nzprotokoli nicht über den Offenbarungsgehalt der e vorgelegt.		
·	• •	ionen dem schriftlichen Sequenzprotokoli entsprechen,		
2. Bestimmte Ansprüche hab	en sich als nicht recherchierbar erw	riesen (slehe Feld i).		
. =	der Erfindung (siehe Feld II).			
A Lineightlich der Paneishnung der E-E-	duna			
Hinsichtlich der Bezeichnung der Erfin Wird der vom Anmelder eing	ereichte Wortlaut genehmigt.			
· =	Behörde wie folgt festgesetzt:			
5. Hinsichtlich der Zusammenfassung [V] wird der vom Anmelder eingereichte Wortlaut genehmigt.				
wird der vom Anmelder eingereichte Wortlaut genehmigt. wurde der Wortlaut nach Regel 38.2b) in der in Feld III angegebenen Fassung von der Behörde festgesetzt. Der Anmelder kann der Behörde innerhalb eines Monats nach dem Datum der Absendung dieses Internationalen Recherchenberichts eine Stellungnahme vorlegen.				
6. Folgende Abbildung der Zeichnungen ist mit der Zusammenfassung zu veröffentlichen: Abb. Nr3				
X wie vom Anmelder vorgesch	lagen	kelne der Abb.		
well der Anmelder selbst kel	ne Abbildung vorgeschlagen hat.			
well diese Abbildung die Erfi	well diese Abbildung die Erfindung besser kennzelchnet.			
				

-		THE R. P. LEWIS CO., LANSING, MICH.	~ ~~~ ~ ***		-	NSTANDES
A .	K LAX	- Z.E.E.U.N.	I OES ARI	MELCILING	344.	NYTANDES

Nach der Internationalen Patentiklassifikation (IPK) oder nach der nationalen Klassifikation und der IPK

B. RECHERCHIERTE GEBIETE

Recherchlerter Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole) $IPK \ 7 \ \ H04Q$

Recherchierte aber nicht zum Mindestprüfstoff gehörende Veröffentlichungen, soweit diese unter die recherchierten Gebiete fallen

Während der Internationalen Recherche konsuttierte elektronische Datenbank (Name der Datenbank und evtl. verwendete Suchbegriffe)

Kategorie°	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
x 🗸	EP 0 817 422 A (TELEFONAKTIEBOLAGET LM ERICSSON) 7. Januar 1998 (1998-01-07) Zusammenfassung Spalte 1, Zeile 22 -Spalte 2, Zeile 50 Spalte 4, Zeile 55 -Spalte 5, Zeile 18	1,10,11, 16,17
Α .	Spalte 7, Zeile 31 -Spalte 8, Zeile 18/	2-9, 12-15

Wettere Veröffentlichungen sind der Fortsetzung von Feld C zu entnehmen	Siehe Anhang Pateriffamilie
 Besondere Kategorien von angegebenen Veröffentlichungen: "A" Veröffentlichung, die den allgemeinen Stand der Technik definiert, aber nicht als besonders bedeutsam anzusehen ist "E" älteres Dokument, das jedoch erst am oder nach dem internationalen Anmeldedatum veröffentlicht worden ist "U Veröffentlichung, die geeignet ist, einen Prioritätsanspruch zweifelhaft erscheinen zu lassen, oder durch die das Veröffentlichungsdatum einer anderen im Recherchenbericht genannten Veröffentlichung belegt werden soll oder die aus einem anderen besonderen Grund angegeben ist (wie ausgeführt) "O" Veröffentlichung, die sich auf eine mündliche Offenbarung, eine Benutzung, eine Ausstellung oder andere Maßnahmen bezieht "P" Veröffentlichung, die vor dem internationalen Anmeldedatum, aber nach dem beanspruchten Prioritätsdatum veröffentlicht worden ist 	erfindedscher Tätickelt beruhend betrechtet werden
The second of the second secon	· · · · · · · · · · · · · · · · · · ·

Datum des Abschlusses der internationalen Recherche	Absendedatum des Internationalen Recherchenberichts
9. Mai 2000	19/05/2000
Name und Postanschrift der Internationalen Recherchenbehörde Europäisches Patentamt, P.B. 5818 Patentiaan 2 NL – 2280 HV Rilswilk	Bevollmächtigter Bediensteter
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Larcinese, C

¢etegorie° '	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Telle	Betr. Anspruch Nr.
A V	HERMANN A ET AL: "PHAMOS - PHILIPS ADVANCED MANAGEMENT AND OPERATIONS SYSTEM - FUNCTIONALITY AND ARCHITECTURE" PHILIPS TELECOMMUNICATION REVIEW, Bd. 51, Nr. 1, 1. März 1993 (1993-03-01), Seiten 30-42, XP000426372 Seite 32, linke Spalte, Zeile 16 -rechte Spalte, Zeile 9 Seite 35, linke Spalte, Zeile 9 - Zeile 43 Seite 38, linke Spalte, Zeile 1 -Seite 39, linke Spalte, Zeile 2 Seite 40, linke Spalte, Zeile 4 -rechte Spalte, Zeile 8	1-17
A	BERGHOLM J O ET AL: "SERVICE DESIGN AND INVENTORY SYSTEM-AN OBJECT-ORIENTED REUSABLE SOFTWARE ASSET" AT & T TECHNICAL JOURNAL, Bd. 75, Nr. 2, 1. April 1996 (1996-04-01), Seiten 47-57, XP000584939 ISSN: 8756-2324 Seite 49, linke Spalte, Zeile 12 -Seite 50, rechte Spalte, Zeile 12 Seite 53, rechte Spalte, Zeile 19 -Seite 54, rechte Spalte, Zeile 8	1-17
,		

•	14	information on patent family members			Inter	national	Application No	- 1
•	intom	ered on patent temby me	mpers		PQ	P	00/02070	ł
Paterit document cited in search report		Publication date		Patent family member(s)	,	. ,—	Publication date	
EP 817422	A	07-01-1998	AU	6842	54	В	11-12-1997	\Box
			AU	49886	93	A	29-03-1994	- 1
			AU	6703	25	В	11-07-1996	
			AU	49888	93	A	29-03-1994	- 1
		•	AU	6926	95	В	11-06-1998	
			AU	52489	96	A	05-09-1996	ŀ
			BR	93056	24	A	01-03-1995	
			BR	93069	83	A	12-01-1999	
			CA	21223		A	17-03-1994	
			CA	21420			17-03-1994	
			CN	10887		-	29-06-1994	
			CN	10883			22-06-1994	
			DE	693228		-	11-02-1999	
			DE	693228			27-05-1999	
			EP	07887			13-08-1997	
			EP	06271			07-12-1994	
	-		ES	21266		T	01-04-1999	
			FI	9508			27-02-1995	
		•	FI	9419			27-04-1994	
			GR	30297		-	30-06-1999	
			JP	85033			09-04-1996	
			JP	75031		T	30-03-1995	
			MX	93051		•	31-01-1995	
			NO	9414			18-04-1994	
			NO	9507		-	26-04-1995	
			SE	4704			11-04-1994	ı
			SE	93003		-	01-03-1994	
			WO	94062			17-03-1994	
			MO	94062			17-03-1994	
			SE	93010			01-03-1994	
			SE	93030			01-03-1994	- 1
			US	56341	.26	A _.	27-05-1997	

PHAMOS – Philips Advanced Management and Operations System – Functionality and architecture | Hoyalalar Hoyalalar Hoyalalar Hoyalalar Hoyalalar

Andreas Hermann, Gerhard Hertlein, Joachim Lenzer, Andreas Lingen, Walter Rothkegel, Norbert Sulzbacher

The Philips Advanced Management and Operations System (PHAMOS) is the general management platform for telecommunications networks. PHAMOS is an open system which is based on international recommendations and industry standards. It supports the distributed management of network elements, equipment and services of heterogeneous networks and the distribution of management functions within a telecommunications management network (TMN). This article describes the functionality and architecture of PHAMOS.

33		OM	Object Manager
Abbreviat	tions	OS	Operations System
		OSF	Operations System Function
API	Application Program Interface	IZO	Open Systems Interconnection
ASN.1	Abstract Syntax Notation 1	OSOM	OS Object Manager
C.ASE	Computer Aided Software Engineering	PDH	Plesiochronous Digital Hierarchy
CMIS	Common Management Information	PHAMOS	Philips Advanced Management and
	Service		Operations System
CMISE	Common Management Information	PMO	Proxy Managed Object
	Service Element	POM	Proxy Object Manager
CMIP ·	Common Management Information	RDBMS	Relational Data Base Management System
	Protocol	SDH	Synchronous Digital Hierarchy
CMOT	Common Management Information	SMA	System Management Application
	Protocol Over TCP/IP	SQL	Standard Query Language
CO	Convenience Object	TCP/IP	Transmission Control/Internet Protocol
DCF	Data Communication Function	TMS	Telecommunications Management
ERM	Entity-Relationship Modelling		System
FTAM	File Transfer and Access Management	WAN	Wide Area Network
FƯ	Functional Unit	WSF	Workstation Function
GDMO	Guidelines for the Definition of Managed	XTI	X/Open Transport Interface
	Objects		
IM	Information Model		
LAN	Local Area Network		
M-API	Management Application Program	l Introdu	ection
	Interface		
MF	Mediation Function	PHAMOS	stands for Philips Advanced Management
MIB	Management Information Base	and Operat	tions System. It is the name of both a deve-
MO	Managed Object	lopment pr	ogramme and a range of products in the area
NE	Network Element	of Networ	k Management and Service Management
NEF	Network Element Function	Systems for telecommunications networ	
NMS	Network Management Function	different t	echnologies, e.g. SDH, PDH, access net-

works, microwave transmission, digital switching and data communications.

PHAMOS provides the general functional and architectural features for a telecommunications management network (TMN) to support the management requirements of Network Operators and Service Suppliers such as:

- > provisioning
- > installation
- > maintenance
- > operation and
- > administration

of telecommunications networks and services [3].

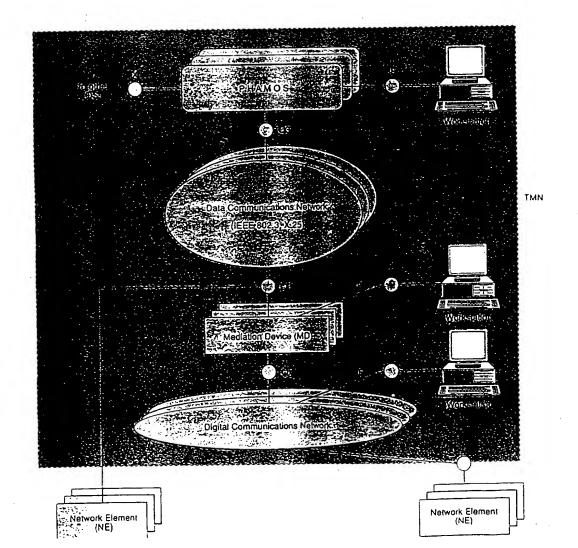
PHAMOS provides management functions for telecommunications networks and services. In this context, a telecommunications network is assumed to consist of network equipment and the associated support equipment. A telecommunications service comprises a range of customised features.

The basic concept behind PHAMOS is to provide a well-organised strategy for the interconnection of

various types of Operations Systems (OS) and/or network equipments with the aim of exchanging management information and initiating respective measures using an agreed system architecture with standardised protocols and interfaces.

The layered structure of PHAMOS consists of a large number of building blocks, implemented in C++, which can be reused for different applications.

In defining the concept, Philips has taken into consideration that almost all network operators, private and public, already have an extensive infrastructure consisting of Operations Systems, networks and telecommunications equipment, which must be integrated into PHAMOS. In this context, it was also recognised that provision must be made for the access to the wide range of management information contained within the TMN and that this information must also be displayed and evaluated. It can be evaluated either by a human, an electronic device or a device of any other type such as alarm displays, alarm bells, video walls, post-processing systems, report generators, expert systems etc.



2 The Telecommunications Management Network

A TMN can vary in size from a very simple connection between PHAMOS and a single telecommunications unit to a very complex network interconnecting many different PHAMOS installations and telecommunications equipments.

A TMN (Fig. 1) provides management functions and offers communication

- > between the PHAMOS installations (these can be NMSs of different kinds) and
- > to the telecommunications network as a whole or between parts of it.

The telecommunications network consists of many types of network equipment, e.g.:

- > terminal multiplexers
- > add/drop multiplexers
- > local cross-connects
- > cross-connects and
- > special units related to the technology used, e.g. repeaters in SDH networks.

Associated equipment such as remote workstations and file servers is also supported. Each managed item of equipment is designated a Network Element (NE).

Such a TMN (Fig. 1) is conceptually a separate network containing various PHAMOS installations which may be functionally different. It interfaces a telecommunications network at various points to exchange information with the network in order to control its operation. A TMN may use parts of the telecommunications network for communication purposes. In SDH the management information is exchanged via the Qecc (Embedded Control Channel). Virtual X.25 connections of the telecommunications network are used in an X.25 net.

2.1 Basic objectives of the TMN

The principle of keeping the TMN logically distinct from the networks and services facilitates the distribution of the TNM functionality for centralised or decentralised PHAMOS implementations. This means that operators can perform the management of a wide range of geographically distributed equipment, networks and services from any of several PHAMOS nodes.

2.2 PHAMOS management functions

The following five management functional areas of PHAMOS can be differentiated in accordance with [3]:

- > Fault management
- > Configuration management
- > Accounting management
- > Performance management and
- > Security management.

A sixth functional area is identified for PHAMOS itself, the System Management Application, i.e. the management of PHAMOS itself.

These functional areas are supported by typical functions such as:

- > Communication interfaces (X, Q) providing for the transmission of information between TMN elements (mainly by using OSI protocols)
- data storage providing for the storage of information over a specific period using the persistent Management Information Base (MIB)
- > retrieval providing access to the MIB
- security mechanisms which control the access to information
- processing mechanisms for analysis and information manipulation, and
- > user interfaces.

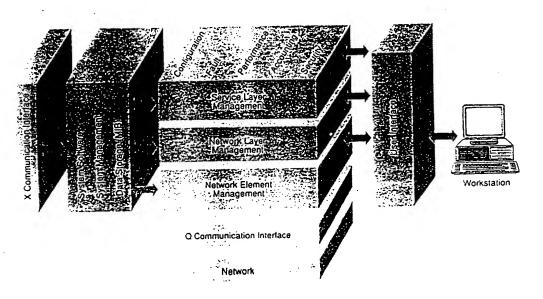


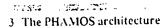
Fig. 2: Functional model of PHAMOS

PHAMOS, as part of a TMN, takes into account that telecommunications networks and services are a defined set of cooperating systems.

2.3 The TMN functional architecture

The TMN provides the means to transport and process information related to the management of telecommunications networks. The TMN functional architecture is based on functional blocks. The functional blocks provide the general TMN functions which enable a TMN to perform the TMN application functions. The TMN functional model [3] comprises functional blocks such as

- > PHAMOS Systems Function for the management of Network Element Functions (NEF); it processes information to support and/or control the realisation of various telecommunication management functions.
- Network Element Functions (NEF): the NEF is a functional block which communicates with the TMN for the purpose of monitoring and/or controlling the NE. This includes all telecommunications and support functions which are required for the management of the telecommunications network and telecommunications functions which are the subject of management.
- > Workstation Functions (WSF)
 which allow the management information user
 to interact remotely with PHAMOS. They provide the means to interpret TMN information.
 They also include a man-machine-interface
 using the graphical user interface (GUI) of
 PHAMOS.
- Data Communication Functions (DCF): these are used for the transfer of information between functional blocks. Their primary role is to provide information transport mechanisms including routing and relaying functions. The standard transport media is a LAN conforming to IEEE 802.3 or CCITT X.25.
- Mediation Functions (MF): mediation functions process information from NEFs and in some cases from QAFs (see below), i.e. they adapt, filter and condense data as required by the PHAMOS installations. The mediation functions can be divided into the following functional components:
 - Information Conversion Function for the conversion of messages. This function characterises systems which have a mediation function and must always be available.
 - Management Application Function which need not necessarily be available. Examples of such functions are temporary storage, filtering, thresholding, data concentration, security, testing, etc.
- Q Adapter Function (QAF): QAFs are used to connect to the TMS those NEFs which do not support standard TMN interfaces. Thus the task of a QAF is to translate between



3.1 Features

PHAMOS – the Philips Advanced Management and Operations System

- > provides mechanisms for customers, value-added service providers and other administrations to access management functions via remotely accessible interfaces of the PHAMOS SW-components;
- > minimises the management reaction time to network events by means of a scalable architecture (Fig. 3) which allows for the distribution of PHAMOS SW-components to multiple servers. The smallest possible configuration is a single workstation whereas the high end configuration consists of multiple servers with various workstations/X-terminals and optional mirrored disks, a "hot-stand-by" server and communication servers:
- > supports the management of any number of network elements, ranging from small configurations to nationwide networks, by the same means as described above;
- > offers mechanisms for distributing management functions within a TMN, e.g. to provide different or identical PHAMOS application services at different locations (PHAMOS installations);
- supports the management of heterogeous networks, equipment and services within the telecommunications environment;
- > provides mechanisms for the interworking between separately managed networks:
- provides mechanisms for technological and functional advances including migration capabilities to enhance early implementations and allow future refinements and extensions.

3.2 Overview of the PHAMOS architecture

3.2.1 The hardware platform

PHAMOS operates on a wide range of hardware configurations ranging from a stand-alone system to control a limited number of network elements to a high availability system including a variable number of servers, workstations, mirrored disks and "hotstandby" servers which are capable of controlling a nationwide network.

The hardware servers, HP 9000 Series 8 x 7 computer systems, used in the current versions of PHAMOS, are based on HP's Precision Architecture RISC (PA-RISC) technology.

I/O slots are available for

- > parallel and serial ports
- > IEEE 802.3 LAN controllers
- > X.25 boards, and
- > hard disk controllers.

HP Series 700 workstations (colour or monochrome)

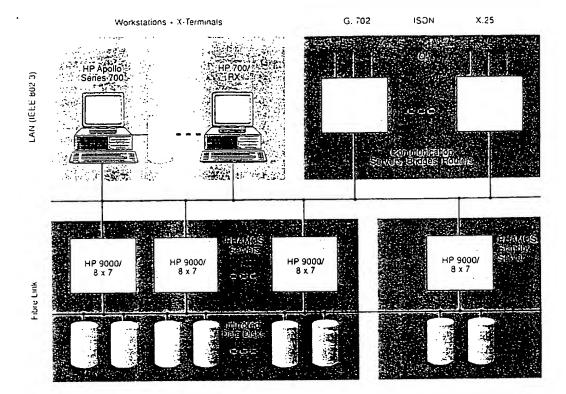


Fig. 3: PHAMOS hardware architecture

functional units of the PHAMOS architecture can therefore be easily distributed.

HP 700/RX X terminals provide a cost-effective solution for the PHAMOS graphical user interface.

3.2.2 The operating system

The UNIX operating system runs on the hardware platform. It complies with AT&T System V.3, the System V-Interface Definition (SVID 2), the X/Open Portability Guide Issue 3 (XPG3), IEEE POSIX 1003.1 and FIPS 151-1. Apart from UNIX, international standards and industry standards such as CCITT Recommendations of the X.200, X.400, X.500, X.600, X.700 [5] series, FTAM, NFS and LU 6.2 are supported.

3.2.3 The network management server OpenView

The generic network management server OpenView is used above the operating system layer. The basic functions of OpenView include:

> Map Hundling

providing mechanisms for editing network topologies (maps), for customising the views of maps, submaps (which can be refined recursively) and network nodes and for connecting applications to map entities. PHAMOS utilises these basic mechanisms programmatically, i.e. the system provides default map configurations (depending

on the functionality of the network elements). These default configurations can be modified by a super-user using the standard editing functions.

> Communication Infrastructure and Object Registration Services

providing basic mechanisms for the communication between managers and agents. These can be located inside PHAMOS or externally in the telecommunications network or in other PHAMOS installations. Due to the location transparency feature of Open-View the PHAMOS managers and agents need not know the physical location of their communication partners.

> Event Sieving;

event sieves are mechanisms by which managers can request the Communication Intrastructure to be notified of certain events only. These events can be generated by Object Managers (OM) with agent functions within PHAMOS (see Section 3.3) – and more relevant – by agents located in the network elements.

3.2.4 OSF/Motif and X-Windows

The following components conforming to industry standards are used for the PHAMOS man-machine interface:

- > X11 Release 4, and
- > OSF/Motif Release 1.1.

The PHAMOS graphical user interface was developed in accordance with the OSF/Motif Style Guide [6] thus facilitating easy introduction to the system.

3.2.5 The data storage system

In the area of management systems the repository of information to be used in management applications is called a Management Information Base (MIB). The MIB of PHAMOS is directly related to the Information Model (IM) of PHAMOS. An IM consists of managed object (MO) class definitions which are described in accordance with the GDMO (Guidelines for the Definition of Managed Objects [1] and ASN.1 (Abstract Syntax Notation 1 [2]). As the MIB is only a conceptual model of all knowledge available in an OS, it is necessary to map it to concrete storage structures either as persistent information in a file or database system or as volatile data within a PHAMOS process. Persistent storage of data in an OS is necessary for the following reasons:

- > The current status and configuration of NEs must be kept to allow for recovery procedures in case of NE failure or outage.
- > Read requests for MOs which are contained in an NE can be cached and therefore more efficiently served. For this purpose, only a database query is required and the need for the resource-consuming communication via CMIS is reduced to a minimum.
- > The information about MOs and services not contained in an NE (OS objects) must be administered in a central repository.
- > An efficient way to address MOs, attributes etc. in the internal communication between PHAMOS components is the mapping of complex identifiers used in CMIP to simple C++ data structures via database query.

For the storage of most of the persistent data a relational database management system (RDBMS) has been chosen, because

- well-defined methods concerning data modelling have been established (e.g. entity-relationship modelling (ERM)) which can easily be transformed into relational storage structures;
- > the semantics and the theory of RDBMS are well defined in the relational calculus:
- > there is a standardised language to access and query a database (Standard Query Language - SQL) which can be easily embedded into normal programming languages like C and C++;
- it is possible to exchange one RDBMS for another without major effort and to communicate between RDBMSs via gateways;
- > they are commercially available and stable and expertise with respect to RDBMS (e.g. tuning) is widely available, and
- it is possible to establish distributed databases which are transparent to the end user.

not easy to map an object-oriented view, provided by the IM, to a relational database scheme.

For this purpose, a set of rules were defined to transform an IM to the first iteration of a data model, more precisely an ERM. This ERM has to be fine tuned by removing redundant information from the IM, taking into account the frequency and the type of access to the data, the performance requirements and other conditions.

In addition, special tables have been defined to allow for the efficient storage of MO identifiers and the very important containment relationship between MOs.

3.2.6 The system management application

Due to its complexity, a PHAMOS system requires a special management application for the management of all components of a PHAMOS installation. This application is called the System Management Application (SMA) which conforms to the general PHAMOS architecture (see Section 3.3) and applies to all management components of PHAMOS. The SMA includes applications with an OSF/Motif user interface for SMA functions and Object Managers with SMA management functionality. The INGRES database is used for persistent data storage.

All entities to be managed are modelled as Managed Objects (MO). Examples of SMA managed objects

- > Elements of PHAMOS and the UNIX operating system: files, disks, printers, backup tapes etc.,
- > PHAMOS processes.
- > hardware components, e.g. physical disks, network interfaces, workstations.

Since PHAMOS is decentrally structured, it must be possible to distribute SMA also. This requirement is fulfilled by partitioning the SMA functionality into three layers:

> User Interface Layer

The user interface layer contains the SMA applications which directly interface to the user. These applications are uniformly integrated into the overall PHAMOS user interface.

> Functional Layer

The functional layer contains the implementations of the SMA management strategies. Example: The functional layer of the SMA security administration component contains the code necessary to define a new PHAMOS user. However, it does not modify UNIX files when making the user known to the operating system.

> Information Access Layer

The object managers of the information access layer contain the proxy objects for system entities such as UNIX files, directory trees, disks etc. When a new network operator is defined, the SMA functional layer accesses the proxy objects for "/etc/passwd", "/etc/ypaliases" etc.

The communication between the SMA components

infrastructure is currently based on CMOT and RCP and will migrate to OSF/DCE in the future. An important aspect of the communications infrastructure is the secure authentication of the communication partners. At the moment, a simple DES algorithm is used to check the credentials supplied by a service requestor. Later PHAMOS releases will employ the RSA algorithm for secure authentication.

The PHAMOS system management application as currently implemented covers the following aspects:

- > Installation and configuration of the platform for the PHAMOS software
 - The platform includes all hardware components and the system software. The SMA supports the initial installation, the upgrades and extensions of the platform and the administration of all relevant configuration parameters.
- > Installation, configuration and administration of the PHAMOS application software The SMA provides user-friendly installation and configuration utilities to manage PHAMOS software releases. This category includes database

software releases. This category includes database management facilities to convert the database contents, if the database model changes with a new PHAMOS software release.

- > Configuration of houskeeping operations
- The SMA provides facilities for automatic backup, log and trace file management. Housekeeping includes mechanisms for the orderly (re-) start and shutdown of workstations and PHAMOS servers.
- > Diagnostic and performance analysis tools These tools are provided by the SMA to assess the current state of the hardware and software components and to predict future shortfalls in performance.
- > Security administration

The SMA provides the complete security framework for PHAMOS. All information relevant to security is kept in a separate security database. The SMA provides a set of convenience objects for accessing data in this database. The PHAMOS security model is based on "access profiles". An access profile defines a set of operations and a set of managed objects on which these operations may be executed.

The management authentication of a PHAMOS user is determined by the set of access profiles which has been assigned to him/her by a security administrator. The security administration contains all the facilities and applications necessary to create, modify and delete PHAMOS users and access profiles and to supervise the state of the security database.

> User authentication

The current PHAMOS version employs a user ID/ password scheme to authenticate a PHAMOS user. In the next PHAMOS release even tighter security mechanisms will be introduced. The user authentication will then be executed using Smart Cards performing an RSA algorithm.

3.2.7 Communication Interfaces

Several classes of interfaces have been defined by international standards bodies (CCITT, ISO, ETSI etc.) to provide for communication between equipment from different suppliers. These interfaces include:

- > Q-interface for the communication between Operations Systems (OS) and Network Elements (NE).
- > Qecc-interface (embedded control channel) using DCCm and DCCr channels for the communication between Network Elements.
- > F-interface for the remote connection of workstations to an OS.
- > X-interface for the interconnection of management systems or Telecommunications Management Networks.

Fig. 4 shows the implementation of the OSI protocol stack in PHAMOS in accordance with CCITT Recommendations G.773, Q.811 and Q.812. The interfaces are accessible by applications via an Application Program Interface (API). At the application layer, two types of protocol profiles are supported:

- > the Common Management Information Service Element (CMISE) which provides transaction type services and
- > the File Transfer and Access Management (FTAM) which provides file transfer type services.

The lower layers support three profiles which are briefly described as follows:

- > A connectionless mode interface using ISO8802-2 type LANs and SCMA/CD (Ethernet) referred to as QB3 in G.773 or as CLNS1 in Q.811,
- an interface using the ISO Internet Protocol over the X.25 protocol (QB2, CLNS2), and
- > a packet interface using X.25 (QB1, CONS1).

PHAMOS can also communicate over other networks, e.g. ISDN, leased lines, SDH payload etc. This is achieved by front ends operating as communication servers.

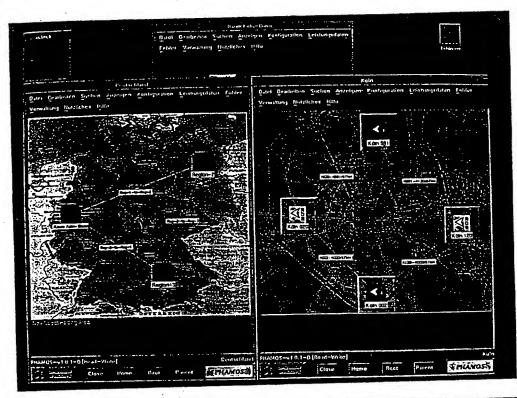
3.3 The Object Oriented Approach

Fig. 5 illustrates the general PHAMOS software architecture. PHAMOS is based on an object-oriented, hierarchical architecture consisting of two abstract entities: applications (Fig. 6) and object managers (Fig. 7). These abstract entities consist of the layers described as follows.

3.3.1 Presentation Layer

The presentation layer exists in applications only. It is used for the visual representation of the entities to be managed and for their manipulation. PHAMOS uses two types of user interface:

- > graphic-based representation via maps and
- > text-based representation via dialogue boxes (Fig. 8).



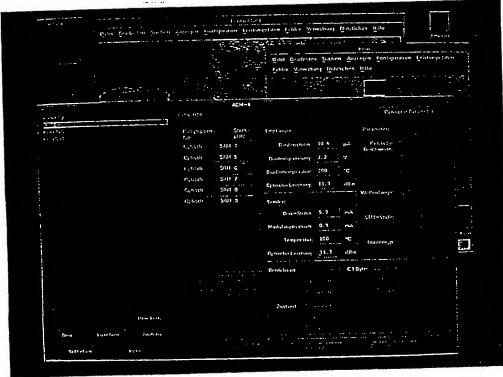


Fig. 8: Examples of the PHAMOS user interface

> Functional Specification

The functional specification describes the functional system behaviour using structured analysis WORK. The information model is developed using GDMO and ASN.1 supported by a proprietary Philips GDMO-oriented tool package.

> High Level Design Specification

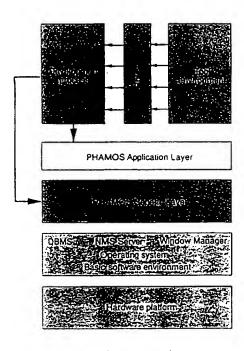


Fig. 9: PHAMOS platform approach

accordance with the generic PHAMOS architecture. The database design, the design of the network topology and the composition of the dialogue boxes are executed in this phase. A prototype is generated using the OSF/Motif User Interface Generator.

> Low Level Design Specification

This specification defines the interfaces of the different PHAMOS building blocks, the message flow between them and the computational structure of their methods and functions. The interfaces of most components are closely related to the information model and are automatically generated by a GDMO tool package.

> Coding and unit test

PHAMOS is written in C++. The module test is suported by a C++ softbench.

is Integration test

A central repository supported by a software management and version control tool from Aide de Camp is used for the system integration. The compilation is controlled by GNU make. The system is integrated incrementally in accordance with the static and dynamic dependencies of the diverse PHAMOS building blocks which are automatically evaluated by analysing the C++ source.

> System test

The system test starts when a PHAMOS system is completely integrated. The system test is performed in the development laboratory following a system test plan. In contrast to the integration test this is a black box test, i.e. the tests are run at the external interfaces of the PHAMOS application.

5 Summary

PHAMOS – the Philips Advanced Management and Operations System – is a universal management system platform based on international standards and industry standards. It provides numerous services for the central or decentral control of the configuration, operation, management and maintenance of large, possibly heterogeneous telecommunications networks. PHAMOS uses parts of the network for its own internal communication with its components. In all other aspects, however, PHAMOS is independent of the network.

References

- [1] Guideline for the Definition of Managed Objects. ISO 10165-4, IS, 8/1991.
- [2] Specification of ASN.1, ISO 8824, IS, 1988.
- [3] Principles for a Telecommunications Management Network, CCITT M.3010, R, 6/1992.
- [4] Reference Model of OSI, CCITT X.200, R, 1984.
- [5] OSI Management Framework Definition, CCITT X.700, R, 10/1991.
- [6] Open Software Foundation. OSF/Motif Style Guide. Revision 1.1. Prentice Hall, 1992.

D. 4/1996 s. 47-57.(11)

Service Design and Inventory System—An Object-Oriented **Reusable Software Asset**

Joseph O. Bergholm J. Michael Davis Behzad Nadji

Peter D. Ting

Competition and the fast pace of technological evolution in today's global telecommunications industry are placing unique demands on the flexibility of operations support software systems. The industry must be capable of rapidly introducing new services, technologies, and organizational structures. Networks must be capable of being partitioned for various applications and administered based on complex ownership relationships among the various network components. User permissions must be readily adaptable to reflect various combinations of services, network partitions, and work functions. To meet the time constraints of the market, telecommunications providers require the ability to configure systems to meet their needs without relying on traditional software development intervals and external software development resources. Traditional software development methodologies generally do not provide the timeliness and flexibility required. The Service Design and Inventory (SDI) system-also known as the Attribute Design Database System (ADDS)-has achieved a high degree of reusability and customer-configurable adaptability through a unique application of object-oriented technology.

Introduction

The Service Design and Inventory (SDI) system is an exceptionally adaptable object-oriented software asset that has achieved the integration of a wide range of functionality and a high degree of software reuse across diverse customer applications. SDI supports several application functions within the network management layer of the telecommunications management network (TMN) architecture (see Figure 1). Within the network management layer, SDI acts as the database of record for all network-related information, including the following:

- Network locational information, including geographic coordinates of structures containing network equipment (customer offices, access serving offices, central offices, and outside plant locations), as well as physical locations of equipment within an office:

- Physical network components, including equipment and transmission media (for example, fiber, coaxial, radio, and microwave), equipment port terminations, and cabling connections within and between equipment; and
- Network path connections, including highline and lower levels of transport facilities and service-providing circuits.

Much of the application functionality within the network design and network inventory management modules of SDI revolves around the management of this highly interrelated network information, including tools supporting network office and bay installation, installation and cabling of equipment and facilities, and circuit design and assembly.

In addition to these network management functions. SDI also serves as the application interface to the service management and element management layers of the TMN. The SDI order management module controls the acceptance and processing of requests for changes in the network originating from planning organizations, as well as customer service organizations. After a request has been processed in the network management layer (for example, a circuit has been designed and assembled), SDI will then communicate the design information to the element management layer to support physical implementation of the design against the appropriate physical network elements. The SDI gateway module supports these types of interfaces to upstream and downstream applications, as well as interfaces to applications in the network management layer for portions of the network that may not be controlled and inventoried via the SDI application.

The final SDI module depicted in Figure 1 is the rule management module. This module allows an SDI customer to tune the application of SDI to support individual processes, services, and technologies. SDI includes

the following customizable aspects:

 Service descriptions, supporting routing, design, and assembly rules for facilities and circuits providing different types of service, as well as acceptable options and settings;

 Processing control schedules and user permissions, allowing a customer to outline the specific control steps and user permissions for various portions of the design and assembly process, as well as permissions management for other modules of SDI:

 Equipment profiles, allowing templates of different equipment configurations to be modeled to support automated equipment configuration, as well as valida-

tion processes:

- Subnetwork definition, allowing various portions of the network to be subdivided based on technology, network hierarchy, organization, or other category to allow separate definition and control of the target subnetworks; and
- Feature (field definition) control, allowing customization of labels, values, help, and error messages to support multiple language delivery.

SDI Development Methodologies

The following subsections detail the four devel-

Panel 1. Abbreviations, Acronyms, and Terms

ADDS—Attribute Design Database System, an alternate name for SDI

CCP—the Customer Care Platform of AT&T Business Communications Services

ITU—International Telecommunications Union

OOA—object-oriented approach

PDH—plesiochronous digital hierarchy

SDH—synchronous digital hierarchy

SDI-Service Design and Inventory system

SONET—synchronous optical network

STM 1—synchronous transfer mode level 1, the STM transport facility with a rate of 155 Mb/s

TMN—telecommunications management network
UIM/X*—User Interface Management System for X
Windows software

opment methodologies of the SDI system: object-oriented analysis, separation of application-specific and core domains, data gateway, and subnetworks.

Object-oriented Analysis Methodology. To support initial analysis and requirements definition activities, members of a team having a wide range of experience with existing network provisioning approaches were assigned to the SDI project. The team members based their initial system specification on an internally generated set of scenarios having the following constraints:

 The specific application domain would be fixed within the network management layer of the TMN.

 The application would handle design and assemble functions for all network hierarchical layers and service types that might potentially be targeted for support.

 The application would handle all existing equipment technologies that might potentially be targeted for

support.

The application would take into account the complex competitive environment in the domain of telecommunications. Finding an entire network that is owned and operated by one telecommunications company or organization is becoming increasingly rare. Thus, the application would have to handle the sharing of inventory use between multiple network management applications.

This open-ended set of constraints forced a dis-

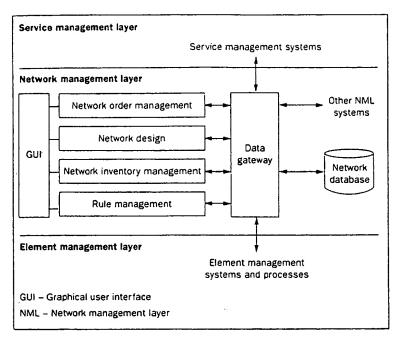


Figure 1. The SDI system is an exceptionally adaptable object-oriented software asset that has achieved the integration of a wide range of functionality and a high degree of software reuse across diverse customer applications. SDI supports several application functions within the network management layer of the telecommunications management network architecture, as shown in the drawing. Within the network management layer. SDI acts as the database of record for all network-related information.

tinctly different approach from that employed for traditional one-off software projects in which development is often based on detailed requirements defined by a single end-user. To maximize software reuse between potential application scenarios, the management of the SDI project adopted the following two design strategies:

- Recognize the similarities between scenarios supporting application functions and engineer around them.
- Separate the similarities found (core aspects) from the application-specific aspects in terms of both data and functionality.

To support this strategy for dissecting the application domain, the team decided to take an objectoriented approach (OOA) to the analysis. In the initial stages of the OOA, a set of primary object classes is typically defined. Then, an inheritance hierarchy is developed based on "typing" information that allows categorization of the class. Initially, typing is rather arbitrary and becomes a sort of classification exercise. True typing does not emerge until scenarios are found that actually result in different attributes and behaviors between different object types.

In the SDI project, the more that was learned about the primary target object classes (link, node,

equipment), the fewer significant differences emerged between the class subtypes. Over the course of the analysis, the class hierarchy became flatter and flatter, or less and less specialized. The team slowly discovered that within the target application domain (network management), differences in object behavior resulted in different relationships between objects rather than fundamental differences in the objects themselves. In other words, one type of object (such as a particular link type) would be defined primarily in terms of the types of objects to which it could be related (node types, other link types). The primary determinants of these relationships were found to be the values of the attributes rather than new or different sets of attributes.

To illustrate this point with the link example, a path or aggregate link can be composed of a number of component links. The allowable types of component links that can constitute this relationship are based on various attribute values on both the aggregate and compositional links (some examples include bandwidth, mileage, restoration priority, transmission media, frame format. and line format). A DS0 circuit, for example, can only be composed of DS1 time slots or other component links meeting the attribute requirements of the DS0 aggregate

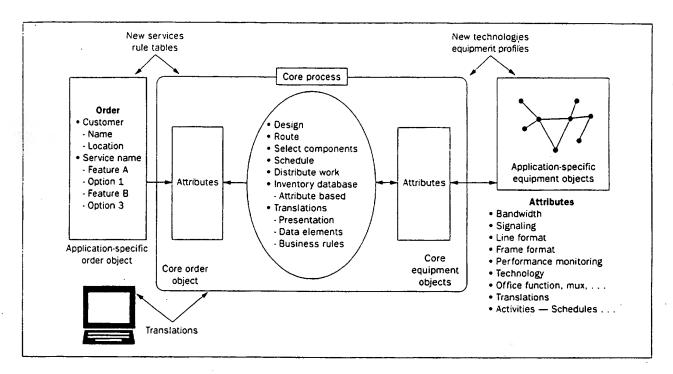


Figure 2. Core objects using application-specific rule data make SDI an adaptable system. The drawing illustrates the separation of application-specific rules from core operations. Using circuit provisioning as an example, the provisioning process can be depicted as being driven by a customer order for a particular type of service, which is defined by various attributes describing that particular type.

link. Although these links differ in the values of their attributes, they all contain the same basic attribute set.

separation of Domains. Based on the results of the preliminary analysis, it was determined that virtually all the basic network management application functionality could be generalized to a set of primary superordinate object classes containing generalized network management operations without the need to develop specialized classes (see Figure 2). Specialization was needed, however, to support the definition of rules for defining which types of objects could be associated with other types of objects in the network. Of course, depending on the network, these rules could differ from application to application based on differences in operations processes,

services, and equipment technologies.

To meet the goal of being able to handle these different application scenarios flexibly, the team had to develop a way to allow the flexible definition of these differences into the application. Toward this end, the team began to develop sets of interrelated rules as data structures in the database that would support object typing (attribute value) information needed to drive the operations supporting the core object classes. Whenever a core operation is triggered, it accesses the appropriate typing information to determine the constraints for establishing relationships to other network objects.

Figure 2 illustrates this separation of applicationspecific rules from core operations. Using circuit provisioning as an example, the provisioning process can be depicted as being driven by a customer order (request) for a particular type of service (type of circuit or link through the network), which is defined by various attributes describing that particular type.

Generally, the service type (or link type definition) is unique to a specific application in both format and jargon. Similarly, each telecommunications provider's

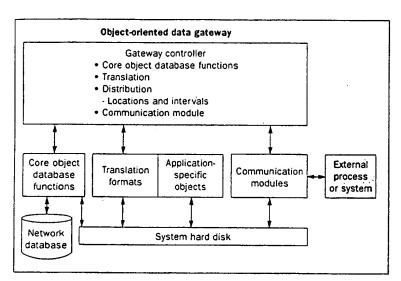


Figure 3. In the SDI data gateway architecture, core data objects, such as service orders, are translated into the formats of application-specific objects either required or supplied by external processes or systems. These external objects are either imported from or exported to the external systems using the appropriate communication module. The gateway supports both multiple translation formats and communication modules.

network is unique in terms of the specific combination of vendor's equipment and design strategies. Thus, the link or service types and network element types can be viewed as application-specific object type definition rules.

Despite the variations in local type definitions, however, all services (as requested by orders) can be viewed in terms of their requirements on the fundamental attributes of a transmission network. These attributes, as illustrated in Figure 2, include such properties as bandwidth, signaling, line format, and technology. In SDI, therefore, the application-specific specialization or typing information is captured separately. Then, it is *instantiated* to core objects prior to processing. Thus, specialization is achieved not through the development of specialized object classes but via the instantiation of specialized attribute values against core, generalized super classes.

The link object class is a good example of using modeling generalization and attributes in the formulation of object classes and objects. Links represent physical or logical connections between two network termination points or equipment ports. They can have inventories (channels) of available capacity. They can have component links at the same level of the network hierarchy (aggregate links) or at different hierarchy levels (which provides a mapping between levels). Finally, links have such attributes as bandwidth, line formats, technology, restoration, and ownership.

By specifying the appropriate attributes under

the control of the SDI rules, link objects can be instantiated representing, for example, any level of the bandwidth hierarchy, any channelization scheme (such as ITU, North American, PDH, SONET, SDH), and any technology (such as fiber, radio, satellite). By appropriately specifying the attributes in the SDI rules, therefore, link objects can be constructed using the same object class code to support the design of a wide range of service types.

This is a radical departure from many traditional system environments. Not only is code not reused, but multiple one-off systems are developed to deal with individual service types, bandwidth levels, or technologies.

SDI applies the same approach to other such generalized object classes as equipment, nodes, and orders. This insures a high degree of code reuse between applications. Furthermore, the rule tables can be defined by users without software development, illustrating the flexibility of the SDI software assets to accommodate changes in a user's environment. It also emphasizes the high level of control the SDI rule application management process places in the hands of a user.

bata Gateway. The SDI data gateway architecture is another example of how the system separates core from application-specific functionality. The gateway structure is designed to provide a flexible methodology for SDI to interface the SDI core capabilities effectively with other application-specific systems and processes. The gateway separates those parts of the code from the

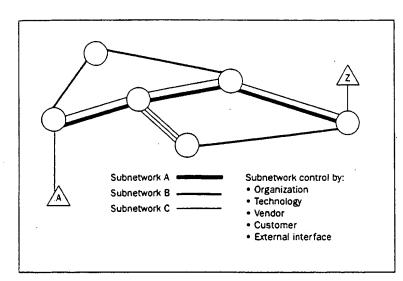


Figure 4. Subnetworks are defined at the link and port levels, and subnetwork attributes are key to management of the network. This drawing illustrates three subnetworks, two of which could represent the responsibilities of organizational regions. The third subnetwork represents a customer's facility between locations A and Z.

core code that may have to change from interface to interface. Such separation helps ensure that the changes can be made in the least complicated way and with the least possible impact on the core portion of the code.

In the SDI gateway as shown in Figure 3, core data objects, such as service orders, are translated into the formats of application-specific objects either required or supplied by external processes or systems. These external objects are either imported from or exported to the external systems using the appropriate communication module. The gateway supports both multiple translation formats and communication modules. This can be as simple as formatting a work order and sending it to a printer or as complex as a full Q3 interface with standards-managed object messages. The gateway controller is a table-driven process that, for a given transaction type, will choose the appropriate SDI core object, translation format, and communication module, as well as the appropriate distribution addresses and frequency (immediate or batch at a given time).

Subnetworks. The SDI network model was designed to allow a network to be divided into multiple subnetworks. This allowed meeting the objective requiring the ability to partition the network to support complex ownership interdependencies in the network management layer. Subnetworks can represent such factors as the network responsibilities of various organizations, the portions of the network leased from various

providers, portions of the network dedicated to the use of a particular internal or external customer, or the portions of the network controlled by various network element controllers. Subnetworks can be defined at a granularity as specific and as low as port or link using tools in the inventory management module. Portions of the network can be "members" of more than one subnetwork.

The subnetwork attribute can be used as a key controlling element in many of the rule tables. When used with the user group and activity tables, the subnetwork attribute can determine, for example, which designer can view (or view and modify) different portions of the network. When the subnetwork is used with the gateway distribution table, it can cause the gateway to send commands to the appropriate network element controller or to send requests for more leased capacity to the appropriate telecommunications provider. With the activity description table, the subnetwork attribute determines such activities as which user groups will be assigned.

Figure 4 illustrates three subnetworks, two of which could represent the responsibilities of organizational regions. The third subnetwork represents a customer's facility between locations A and Z.

Figure 5 illustrates the application of subnetworks to identify a portion of the network obtained from an access provider between nodes A and D and the establishment of subnetworks at specific network hierarchy levels. For example, the STM 1 SDH network between

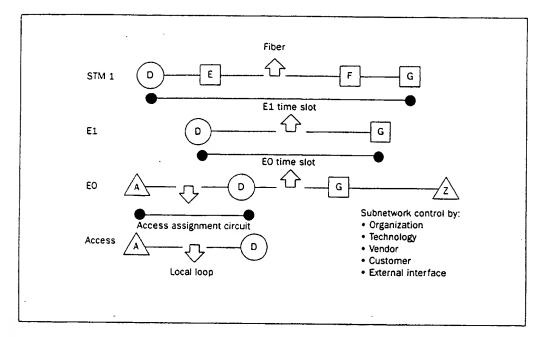


Figure 5. Subnetworks are applied to various levels of the network hierarchy. This illustration shows the application of subnetworks to identify a portion of the network obtained from an access provider between nodes A and D and the establishment of subnetworks at specific network hierarchy levels. The STM 1 SDH network between locations D and G may be the responsibility of a specific internal organization.

locations D and G may be the responsibility of a specific internal organization.

The use of the subnetwork attribute in the network definition and in the rule tables illustrates another capability that SDI places in the hands of a user to adapt the SDI software assets to changes in the business environment.

SDI Software and Hardware Architecture

SDI employs a logically three-tiered client/server architecture as depicted in Figure 6. Tier 1 normally consists of a workstation containing the user interface, some functional modules, and the database interface layers. Tier 2 consists of a local file server for the Tier-1 workstations and printers in the same cluster. The database and its server functions are located on a single Tier-3 server, also known as the *central server*.

Depending on the customer and the customer's geographical distribution of offices, two or three of the tiers can physically collapse into one. Thus, as shown in Figure 6, SDI can range from fully residing on a single workstation to a two-tiered client/server model on a local area network to a three-tiered architecture on a wide area network. This architecture is fully aligned with the AT&T

Business Communications Services (AT&T-BCS) Customer Care Platform or CCP, which is currently globally deployed.

The software architecture of SDI, as shown in Figure 7, was designed using multiple layers. Where appropriate, an object-oriented approach was designed within layers to ensure ease of maintenance and porting, as well as data encapsulation.

The physical user-interface layer provides direct graphical interface to a user. It separates graphical user interface toolkit and platform dependence from the rest of the application. This strategy proved useful as SDI was ported from the OPEN LOOK* to the Motif* platform, only this layer had to be translated. This means that the switch to Motif* was made without affecting code other than the code at the physical UI layer. A third-party tool, the User Interface Management System for X Windows software (UIM/X*), is used to design the physical layer. The user-interface library layer, which functions as a set of C++ class objects and methods, is an object-oriented implementation toolkit (an independent window presentation). The control logic that drives the physical window layer through user-interface library objects resides in the abstract user-interface layer. Operations that manage the

TRETTECUMENT INFORMAT - MANCHARDS INC. .

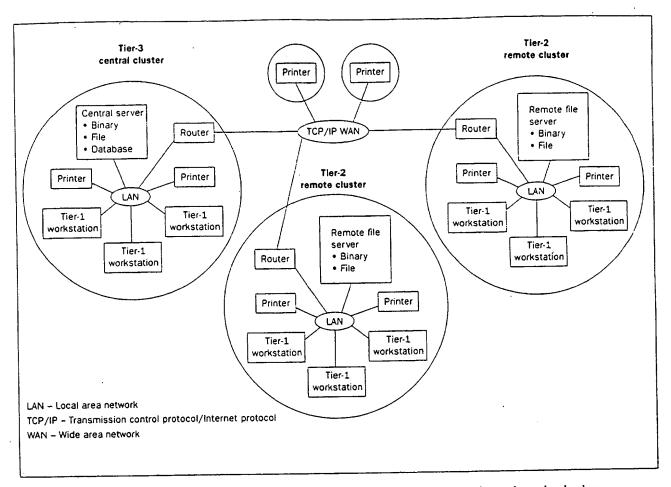


Figure 6. SDI employs a logically three-tiered client/server architecture as depicted in the drawing. Tier 1 normally consists of a workstation containing the user interface, some functional modules, and the database interface layers. Tier 2 consists of a local file server for the Tier-1 workstations and printers in the same cluster. The database and its server functions are located on a single Tier-3 server, also known as the central server.

physical window (such as zoom in/out and enable/disable buttons) are implemented in this layer. Finally, application-layer objects requiring visual presentation are grouped in the presentation layer.

The software architecture that maps the object-based modules into the relational database is illustrated in Figure 8. This drawing also depicts key characteristics of the SDI software architecture, as well as aspects of the modeling and development tools employed. The functional object-oriented modules¹, programmed in C++, interface to the ORACLE7* relational database management system through an object-oriented database interface layer, which is code generated using the specification-driven code generator MetaToolTM. This layer provides the func-

tional modules' independence from the database.

SDI object modeling drove the definition of the table structures in the relational database. Using Rumbaugh's techniques¹, the tables closely resemble the objects themselves. Denormalizing some tables in the database leads to improved performance for the database calls. Tuning the database for performance has been an important project activity.

To further improve performance and to separate the functional code from the details of the database realization, stored programs within the ORACLE system have also been employed. This is particularly important in light of the scaleable SDI hardware architecture depicted in Figure 6 in which the stored procedures reduce the amount of data that must be transmitted on the wide area networks between the central server and the clients.

SDI uses several third-party tools and libraries in its design. AT&T standard components libraries are a key part of SDI design. In addition, the use of AT&T BaseWorXTM system components, third-party map and table widgets, as well as hypertext have helped reduce the time required for development of the SDI system.

ATETTOCINICAL INCOMAL ANADOM/APPIL TOM

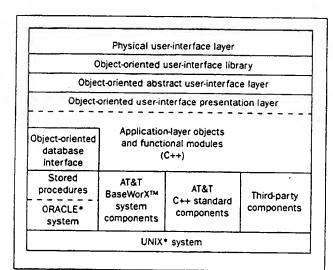


Figure 7. The software architecture of SDI, as shown in the chart, was designed using multiple layers. Where appropriate, an object-oriented approach was designed within layers to ensure ease of maintenance and porting, as well as data encapsulation.

SDI Deployment Activities

The deployment of the SDI system is based on a particular software release and includes the following three major activities:

- Characterization of the user's environment to obtain the information necessary to populate the appropriate tables and rules, ensuring that the core components will meet environmental needs;
- Assistance with the initial database population; and
- Identification and development of the necessary application-specific components.

With the rule based structure of the SDI system, the capabilities required by a customer are partially realized both by the functionality of the code and by the populated rules. System testing of the code releases, therefore, only partially ensures that the customer's requirements will be realized.

Thus, a dual-cycle testing strategy has evolved from the SDI project. This strategy consists of through-system code testing followed by testing of the rules within the environment of the tested code. The rule testing generally takes the form of a regression of a key

subset of the system tests combined with scenario testing based on specific customer application processes. After the entire testing cycle is completed, the system is deployed having an initial set of customer rules.

Rule management tools are available to enable an application administrator to modify the tables and rules to accommodate most changes in their environment, such as new services, technologies, and organizational responsibilities. To ensure the integrity of the rule data, however, the administrator must establish a test data environment for basic scenario testing before the introduction of new data structures into the operational environment.

Specific Deployment Activities. Currently, SDI has been deployed to four customers, including those in both the Spanish and English environments. The first four applications represent a wide variety of network configurations, service types, organizational structures, and work flows. In addition, the following basic technologies addressed in the customer networks varied widely:

- Transport capacity design and leased line circuit design for a network based on E5 fiber and E4 radio technologies:
- Trunk networks between customer PBXs, access switches, and backbone switches having leased transport at the E1, E3, or E4 bandwidth levels; and
- Transport capacity design involving an SDH-based backbone network initially supporting a switched trunk network but evolving to a broad set of voice and data services.

In addition to these fundamental differences in deployed technologies, the operations processes and user permissions management have also differed widely between the initial SDI customers. Despite the diverse nature of these applications, SDI has achieved virtually 100-percent software reuse across the deployments, discounting the software enhancements between releases for a product early in its development cycle. These achievements are a direct result of the design objectives adopted at the beginning of the project.

Conclusion

The definition of generalized, attribute-based object classes in SDI enables the same code to be applied to a wide range of user scenarios. To operate, these generalized object classes are specialized via the instantiation of attributes retrieved from user-populated rule tables.

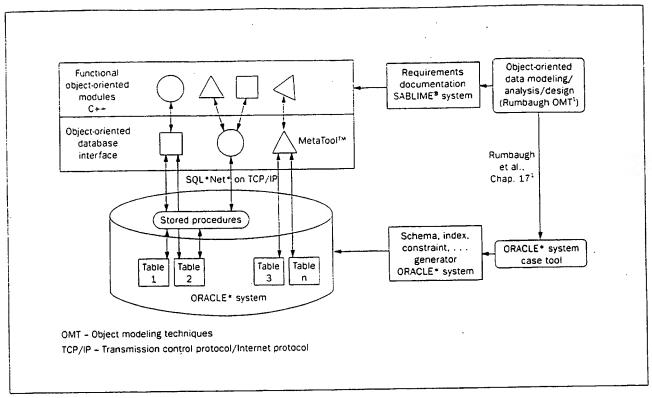


Figure 8. This drawing shows the mapping of objects in SDI to relational database tables. It also depicts key characteristics of the SDI software architecture, as well as aspects of the modeling and development tools employed. The functional object-oriented modules, programmed in C++, interface to

Thus, not only is the same code reused, but the operation of the system can be configured by a user without additional code development. This arrangement reduces the time and expenditure necessary to customize SDI to meet application requirements, thereby reducing the time and cost required to introduce new technologies. services, and user-defined processes and permissions.

The adaptability of the object-oriented attribute and rule-based software has many user advantages in today's competitive and rapidly evolving telecommunications environment. The maintenance and generation of quality rules, however, is a responsibility that requires a thorough understanding of the user operational environment and SDI rule structures, as well as the discipline to test and verify the operation of the rules as the environmental needs evolve.

Acknowledgments

Many people have contributed to SDI. The leadership and vision of S. Y. Lee resulted in the initiation and support of this effort. Support from AT&T-BCS globalization fund providers Ed Sable and Don Wormley was instrumental in SDI's evolution in recent years. Prem

the ORACLE7* relational database management system through an object-oriented database interface layer, which is code generated using the specification-driven code generator MetaTool™. This layer provides the functional modules' independence from the database.

Mehrotra, Debra Williams. Mohammed Mehio, Hai-Chen Tu. Joe Russell. Steve Lessard. Cheryl Histand. and Rod Pease were involved since SDI's initial design and architecture development. Currently, a team of nearly 50 dedicated people from AT&T-BCS and AT&T Network Systems are actively contributing to SDI engineering. development. testing, customer relations, and support.

*Trademarks

Motif is a registered trademark of Hewlett-Packard Inc.
OPEN LOOK is a registered trademark of Sun Microsystems Inc.
ORACLE, ORACLE7, and SQL*Net are registered trademarks of
ORACLE Corp.

UIM/X is a registered trademark of Visual Edge Software, Ltd.

Reference

 J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. Object-Oriented Modeling And Design, Prentice Hall. Englewood Cliffs, New Jersey, 1991.

(Manuscript approved February 1996)

Joseph O. Bergholm is a technical manager in the



Globalization, Provisioning, and Technology Planning and Systems Department at AT&T Business Communications Systems in Middletown, New Jersey. Mr. Bergholm is responsible for planning, systems engineering, and

customer deployment management in support of the Service Design and Inventory System (SDI). He was graduated from the University of Pennsylvania in Philadelphia with B.S. and M.S. degrees in electrical engineering, as well as a Ph.D. in systems engineering and operations research. Mr. Bergholm, a recipient of the AT&T Eagle Award for SDI system technology and a patent holder in digital transmission technology, joined the company in 1966.

J. Michael Davis is a district manager in the Globalization,



Provisioning, and Technology Planning and Systems Department at AT&T Business Communications Services in Middletown, New Jersey. He manages and supervises all Service Design and Inventory System (SDI) engineering

processes. Mr. Davis, who joined AT&T in 1988, has a B.S. degree from Miami University in Oxford, Ohio. He also has an M.S. degree in cognitive science (artificial intelligence) from the University of Cincinnati in Ohio, where he is also a doctoral candidate.

Behzad Nadjl is a technical manager in the Globalization,



Provisioning, and Technology Planning and Systems Department at AT&T Business Communications Services in Middletown, New Jersey. He is responsible for the development of network inventory and provisioning systems. He is

currently working on the Service Design and Inventory (SDI) system, as well as other system development projects. Mr. Nadji has an M.S. in electrical engineering from the Electronics University of Tehran in Iran. He also has M.S., E.E.E. (engineer's), and Ph.D. degrees in electrical engineering from the University of Southern California in Los Angeles. He joined AT&T in:1988.

Peter D. Ting is a technical manager in the Provisioning



Systems Development Department at AT&T Business Communications Services in Middletown, New Jersey. He manages an organization responsible for designing and developing software systems that automate telecommunications service provi-

sioning processes. He has a B.S. degree from the National Taiwan University in Taipei, an M.S. from Brown University in Providence, Rhode Island, and a Ph.D. from the University of California at Berkeley, all in physics. Mr. Ting, an AT&T Bell Laboratories Fellow, joined the company in 1973.



Europäisches (

tamt

European Patent Office

Office européen des brevets



(11) EP 0 817 422 A2

(12)

EUROPEAN PATENT APPLICATION

(43) Date of publication: 07.01.1998 Bulletin 1998/02

(51) Int. Cl.6: H04L 12/24

(21) Application number: 97117274.7

(22) Date of filing: 18.08.1993

(84) Designated Contracting States:

BE CH DE DK ES FR GB GR IE IT LI NL

(30) Priority: 28.08.1992 SE 9202488 05.02.1993 SE 9300363

(62) Document number(s) of the earlier application(s) in accordance with Art. 76 EPC: 93919758.8 / 0 627 143

(71) Applicant: TELEFONAKTIEBOLAGET LM ERICSSON 126 25 Stockholm (SE)

(72) Inventors:

- Carebrand, Per-Arne 118 42 Stockholm (SE)
- Svedberg, Johan
 115 24 Stockholm (SE)
- Fantenberg, Johan
 112 47 Stockholm (SE)

- Talldal, Björn 161 46 Bromma (SE)
- Palsson, Martin
 146 45 Tullinge (SE)
- Gilander, Anders
 195 54 Märsta (SE)
- Sellstedt, Patrik
 124 71 Bandhagen (SE)
- Strömberg Stefan
 141 52 Huddinge (SE)
- (74) Representative:
 Rosenquist, Per Olof et al
 Bergenstrahle & Lindvall AB,
 P.O. Box 17704
 118 93 Stockholm (SE)

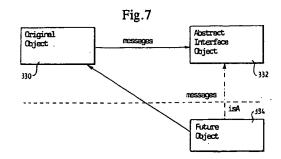
Remarks:

This application was filed on 06 - 10 - 1997 as a divisional application to the application mentioned under INID code 62.

(54) A method for implementing a managed object in a network management system

(57) The invention relates to a method for implementing a managed object in a subsystem of a managed system in a management network with at least one managing system and at least one managed system, for telecom or open systems, said managed system consisting of subsystems forming each a part of the managed system including one or more managed objects.

The managed objects are implemented in the subsystem, uncoordinatedly with respect to other subsystems, in such a way that they can be connected to and transmit messages to other objects in other subsystems, and without knowing the type of objects in the other subsystems. This is obtained by designing a first object (330) for co-operation with an abstract object (332) defining an interface consisting of unimplemented methods, which may be called by the first object, and letting, at later design of a second object (334) unknown to said first object and intended to co-operate with the first object, the other object inherit the abstract object and implement the inherited methods, so that the first object at co-operation with the second object will consider it as being of said abstract type.



EP 0 817 422 A2

Description

Technical field

The present invention relates to a method for implementing a managed object in a management network with at least one managing system and at least one managed system, for telecom or open systems, said managed system consisting of subsystems forming each a part of the managed system including one or more managed objects.

With a "management network with at least one managing system and at least one managed system" is meant that the management network can include at least one managing system which can manage one or more managed systems, which likewise can form part of the management network.

With an open system is meant a system of the kind, which is defined in Reference Model of Open Systems Interconnection (OSI) for CCITT Applications, Rec. X.200.

To perform management activities in a management domain there must be at least one manager which is responsible for the management of the resources. A resource is something which includes concepts and ability of the domain. An example of a domain is a telecom network, where the resources are switches, trunks etc. and management units are e.g. operator tools and managing systems for the network.

For operation of telephone networks each individual company has used a number of different systems for operation and maintenance. CCITT has developed a standard model for operation and maintenance in telephone networks, called TMN (Telecommunication Management Networks). The basic principle of TMN is to indicate an organized network structure, which admits connection of various managing systems to telecom equipment. This is achieved by use of standardized protocols and interfaces. The telephone companies and other operators will require that future telecom networks are adapted to TMN.

CCITT has a recommendation for this under development, the M.3000-serie.

TMN considers all network nodes as network elements (NE). These network elements are made as telephone switches and transmission or transport network products.

The functional structure of TMN comprises

- management functions (OSF, Operations Support Functions), which manage application programs available for users, such as management functions for "Business Management" and service and network administration;
- data communication functions (DCF; Data Communications Functions), which manage data communication between the managing systems OSS and the managed systems NE;

 mediation functions (MF, Mediation Functions), which convert information (between e.g. managed objects), manage data, concentrate, reduce and edit, make decisions relating to e.g. threshold limits and store data, which identify equipment and networks;

2

- network element functions (NEF, Network Element Functions), which manage telecom processes as switching functions and transmission, and take part in management processes for telecommunication, as fault localisation and protection connections;
- functions for interface adaption (QAF, Q-Adapter Functions), which perform conversion of interfaces from non standard to standard;
- work station functions (WSF, Work Station Functions), which constitute the user terminals of TMN, show information and assist management technicians to manage the network.

TMN also includes an interface, called Q3. Q3 is besides being a communication protocol also an information model, which comprises data schemes, operations and notifications. The exact details of the Q3 interface and its protocols appear from the CCITT recommendations Q961 and Q962.

TMN considers all physical and logical objects as managed objects, which in TMN are referred to as MO (Managed Objects), a denomination which will be used alternately also here henceforth. Managed objects are data images of such physical or logical resources as wires, circuits, signal terminals, transmission routes, events logs, alarm reports etc...

A specific relationship occurs between a resource and a managed object. A resource can be related to one or more MO or none at all. On the other hand an MO can be related to one or more resources or none at all. This relationship is important if an MO is affected by some form of operation or maintenance activity. An MO must not be removed before the functions to which it is subordinated have themselves been removed. This information model is based upon object orientation and the relation concept.

The managing system (OSS- Operation Support system) treats network elements and subordinated managing systems as a collection of managed objects in an imagined data base MIB (Management Information Base). These managed objects are constituted by instances of an MO class, such as a number of signal terminals of the same type. Each terminal will thus constitute an instance of the class signal terminals.

In TMN there also exists the concept MIM (Management Information Model), which refers collectively to all information related to managed objects. MIM is a model of all attributes, relations, operations and notifications referable to managed objects. To be able to search for MO-instances a management information tree MIT (Management Information Tree) is used. This tree structure starts in the network and indicates network ele-

ments and subscribers or equipment.

For operation and maintenance each separate unit or resource, about which information is required, or the operation of which is influenced from the outside of the system, is represented by a managed object. Each information exchange referable to the management of operations or units, which must be influenced or which must report something (such as set up of data, assignment of name, or management of alarms) is done in the form of operations on, or notes from managed objects.

A managed object includes attributes and can also have a relation to other managed objects. A number of different operations can be directed towards a managed object and events can be generated by such objects.

Below an account of deficiencies of TMN will be given.

Network elements and subordinated managing systems are managed by a managing system by means of operations towards the managed objects in the managed systems and supervision of notifications transmitted by the managed system.

Allowed operations towards managed objects in the managed system are determined by an information model of the managed system, together with the notifications, which can be transmitted. The information 25 model states:

- which classes of managed objects that are defined,
- which operations the managed objects in these classes accept,
- which notifications that the managed objects are expected to transmit,
- how many instances that can be created or removed, in each class of managed objects,
- dependences between managed objects, e.g. that one managed object requires existence of another
- dependences in a managed object, e.g. that a specific attribute value of one attribute is only allowed if another attribute is set to a specific value or if the managed object only can be removed if it is in a specific state.
- the purpose and intention with managed objects and their notifications.

To be meaningful the managing system must know the information model of the managed system. This is in TMN called "shared management knowledge".

At a change of the information model the management model must be updated in accordance with this change. In conventional systems these changes are made by:

 Definition of the new information model. This is made by specifications for managed objects written as GDMO/ASN.1 templates (GDMO - Guidelines for the Definition of Managed Objects according to CCITT rec. X.722 ISO/IEC 10165-4) and ER-diagrams (Entity-Relationship diagrams) for the managed objects. The specifications for the managed objects state formally (machine readable) syntax (e.g. operations and notifications) for the managed object.

All other parts of the information model, as dependences, the number of instances etc., are stated informally as comments in natural language.

- Implementation and verification of the new information model in the managing system and the managed system.
- Confirmation that the managing system and the managed systems are adapted to the same information model by performing accepted test sequences.
- Updating of the network consisting of the managing system and the managed system with this new version of the information model.

That just mentioned above results in a number of problems:

Firstly, the development of managing systems and managed systems must be coordinated, which leads to higher development costs and/or a delayed introduction of new services on the market.

Secondly, the absence of formalism with respect to the specifications of the managed systems, makes implementation, verification and acceptance of both managing system and managed systems to a difficult and time demanding task, since the interpretation of the specifications is open to discussion.

Thirdly, updating of networks must be planned and carried out carefully, as there are dependences between different versions of the managing systems and managed systems. This involves a delayed introduction of new services in the network.

The purpose of management according to the TMN model is to be a framework for standardization of management for telecom networks and open systems. The management structure strongly influences the management paradigm for new system architectures. There are strong reasons to assume the management paradigm according to the TMN model for the whole management and not only for domains subjected to standardization. The main reason for this is that it is desirable to be able to develop and design management functions in a uniform way.

Summary of the invention

It is a purpose of the invention to design a managed object, which should be able to communicate with unknown future objects. It should be possible for the original object to be related to and to co-operate with an

15

45

55

unknown object without modification, or even reloading of the original object.

Said purpose is attained by the method defined by way of introduction being characterized in that the managed objects are implemented in the subsystem, uncoordinatedly with respect to other subsystems, in a way that they can be connected to and transmit messages to other objects in other subsystems, and without knowing the type of objects in the other subsystems, by designing a first object for co-operation with an abstract object defining an interface consisting of unimplemented methods, which may be called by the first object, and having, at later design of a second object unknown to said first object and intended to co-operate with the first object, the second object to inherit the abstract object and implement the inherited methods, so that the first object at co-operation with the second object will consider it as being of said abstract type.

Advantageous embodiments of the invention have obtained the characterizing features stated in the 20 respective dependent claims.

Description of the drawings.

A number of embodiments of the invention will now be described below in greater detail with reference to the enclosed drawings, on which

Figures 1-9 in block diagrams illustrate problems. which can appear at the use of technology according to the state of the art in connection with the design of managing systems, where

Figures 1-4 concern problems, which can appear in connection with reuse of library components in the managed objects in managed systems,

Figures 5 and 6 refer to problems, which can appear at the implementation of a managed system in a layered system architecture,

Figures 7-10 illustrate how the problems according to Figures 1-6 can be generally solved by design of a specific type of object, which should be able to cooperate with unknown future managed objects.

Figures 11-14 specify design of objects according to Figures 10-14, the object type being assumed to belong to a platform system,

Figures 15-20 show the implementation in the program code C++ of dependences between the designed objects according to Figures 11-14.

Description of embodiments

A managed system consists of several subsystems, more exactly a system platform and a number of applications. According to the invention these subsystems are developed uncoordinatedly and are loaded sepa-

At the development of an application - or a system platform - there are libraries with reusable components.

These components should be incorporated and combined in different ways in the different subsystems.

The two different problems described above have certain properties in common. There are system components, which are developed uncoordinatedly, but still there are dependences between the components. To keep such dependences it is necessary that the components should be able to communicate with other components which are separately developed and even loaded.

The design process described below in greater detail is usable in two different contexts with similar problems.

The first problem refers to reuse of library components.

At design of a managed system there are libraries with reusable components, which can be incorporated in the objects of the managed system. As is illustrated in Figure 1 there are first a main library 260 including components with basic functionality, such as MOstateComponent 262 and WorkingStateComponent 264. These components include state attributes common to many different types of managed objects. The functionality of these basic components is reused by other components with a more specialized functionality. In the traffic management library 266 there is e.g. a StatePropagation-Component 268, which has the ability to transfer state transitions in MOstateComponent 262 to the related dependent object 270.

It should be emphasized that components as well as the managed objects are supposed to be implemented in an object oriented language. Both the components and the managed objects are thus really objects. But the components can not exist as identifiable objects by themselves in an application. They can only form a part of managed objects.

A straightforward way to implement components, which depend on other basic components, e.g. the components in the fault handling and the traffic management libraries in Figure 1, is to include the basic components either by heritage or aggregating. This causes however many problems. This depends upon the fact that several components, e.g. FaultHandling-Component 272 and ResourceManagementComponent 274, which inherit or aggregate the same basic component, e.g. MOstateComponent 262, can be included in the same managed object-Route 276, compare Figure 2. This implies that there will be several copies of the basic component in the managed object. which will cause consistency problems if the basic component includes data, which is visible external to the components. Data of MOstateComponent should in other words be available for another functionality in the managed object than those components in which MOstateComponent is included.

There is a way to avoid the problem described above.

MOstateComponent should not be inherited or aggregated in components which depend on this component. Instead the dependent components should include reference attributes to MOstateComponent. Thus the latter will be a component of its own in those objects in which it is included. Components which need to access MOstateComponent will then each include a reference to the same instance of the MO-state component, compare Figure 3.

As has been mentioned earlier a component, such as ResourceManagementComponent 274 and Fault-HandlingComponent 272, can include a functionality which is triggered by state changes in MOstateComponent 262. The latter must therefore be able to transmit state change messages to other components. One problem is however that the receivers of these messages are unknown at the design of MOstateComponent. MOstateComponent 262 must in some way be able to communicate, cf. arrows 280, 282, with an arbitrary later appearing component, compare Figure 4.

The second problem refers to a layered system architecture.

A managed system can be implemented in a layered architecture. More exactly, there are system platforms with basic functions, which can be reused by different applications, compare Figure 5. The system platform 290 i.a. includes different kinds of common telecom services. The application can thus delegate many tasks to the system platform. The system platform 290 and the applications 292 are loaded separately. It must be possible to load an application and link it to the system platform without reloading the platform.

Both the system platform 290 and the applications 292 include managed objects. The objects 294 and 296 in the platform 290 can e.g. represent resources, as switches, transmission routes, trunks etc. The objects 298,300 and 302,304 in the applications 292.1 resp 292.2 can be related to and co-operate with the resources in the system platform, compare Figure 6. An object in an application can delegate some tasks to an object in the system platform. Objects in the applications may thus depend on related objects in the system platform to be able to work. As the system platform is known at the development of an application it is no problem to design the objects in the applications for communicating with the objects in the system platform. As has been mentioned earlier the objects in the applications can however be dependent on the objects in the system platform. This implies that objects in the system platform should be able to transfer state changes, as has been described earlier above, to objects in the applications. This implies that objects in the platform should be able to be related to and take initiative to calling objects, which are unknown at the design of the platform system.

Above two cases with similar problems have been described. Here will now follow a description of how these problems can be solved.

More exactly, the real problem in each of the two cases is to design an object, which should be able to communicate with unknown future objects. It must be possible for the original object to be related to and to cooperate with an unknown object without modification, or even reloading of the original object.

This problem can be solved with the design principle illustrated in Figure 7. The original object 330 is designed to co-operate with an abstract object 332. The abstract object defines an interface consisting of unimplemented methods. These are the methods, which can be called by the original object. At the design of an object 334 designated to co-operate with the original object it must inherit the abstract object 332 and implement its inherited methods. At the co-operation with an unknown type of object the original object 330 will consider this as being said abstract type 332. At call of a method defined in the interface of the abstract object 332 the call will be delegated to implementation in the real object 330 via late binding.

Avoidance of reloading of the original object at loading of the future objects is achieved by dynamic linking.

Figure 8 illustrates how the just described design can be used for the design of basic library components. MOstateComponent 340 is designed to communicate with objects, which inherit an abstract object: MOstateSubscriber 342. This abstract object defines methods called by MOstateComponent when a state change occurs. If a component should subscribe to state changes in MOstateComponent, it must inherit MOstateSubscriber 342 and implement the answer to the respective state transition notifications. The subscribing component must of course also state itself as a subscriber to MOstateComponent.

300

The principles of design described with reference to Figure 7 can also be used for designing objects in a system platform to co-operate with application specific objects. This is illustrated in Figure 9, where object PlatformObject1 350 is designed to co-operate with object 352 in application system 354. An object in an application, which should co-operate with the object Platformobject1 350 must inherit the abstract object Interfaceobject 356. Further there is a data base relationship between PlatformObject1 350 and InterfaceObject 356. This implies that the objects which inherit InterfaceObject 356 can set up relations with PlatformObject1 350. To make it possible to load applications without reloading of the system platform dynamic linking is used.

Often an object in an application system is state dependent of another object in the system platform. This design makes it possible to implement such state dependences in essentially the same way as for objects which are known to each other when they are implemented, as will be described in greater detail further below.

Dependences between uncoordinated objects can be specified and implemented in essentially the same way as between coordinated objects, which has been described earlier above. Here the same examples will

45

50

be used to show the design and implementation of dependences between uncoordinated objects. It is however assumed here that the object type ResourceB belongs to a platform system. Thus, with reference to Figure 10, various kinds of objects in various application systems can be related to and co-operate with the object ResourceB 370. The object type ResourceB 370 is related to UserObject 372 via a data base relationship. ResourceA 374 inherits UserObject 372 to be able to be related to ResourceB 370. In the following it will be shown how these objects are specified and implemented. The same pseudo syntax as above will be used

The specification of the object ResourceB is found in Figure 11. The specification includes two state attributes: admState and opState, a method to allocate instances of ResourceB and a precondition which states that to enable erase of a ResourceB object it must be locked. This is essentially the same as earlier. The only difference is that in this example ResourceB is related to UserObject instead of ResourceA. The relation is established via a reference attribute userRef, line 6.

In Figure 12 the object UserObject is specified. It includes the state attribute admState. It can however be noted that this attribute is specified as being purely virtual, line 19. This implies that the UserObject in fact will not include the attribute admState. The interface of UserObject will however include unimplemented write and read methods for this attribute. The real implementation of these virtual methods will appear in the respective subtypes of UserObject. Besides there are another attribute resourceBstate, which is assumed to keep the value of opState in the related object ResourceB. Further, the UserObject has a reference attribute Bref, which is used to establish relations with objects of the type resourceB, line 21.

A dependence scheme, which specifies end conditions, including both the object UserObject and the object ResourceB is found in Figure 13. There is an end condition, lines 29-31, which states that admState of UserObject depends on admState in ResourceB in such a way that an object UserObject can not be locked up if the related object ResourceB is locked.

In Figure 13 it is further specified that when a value of opState in an object ResourceB is changed, the new value should be propagated to the related instance of UserObject. In Figure 13 there is no end condition specified which relates to opState, it is only stated that changes of opState in ResourceB should be propagated to the attribute resourceBstate in UserObject, lines 36 and 37. This implies that all subtypes of UserObject are not necessarily opState dependent on ResourceB, but each subtype of UserObject can manage the opState dependence of ResourceB in its own way.

In Figure 14 the specification of ResourceA is shown. ResourceA is specified as being a subtype of

UserObject (line 41). This implies that ResourceA inherits the attributes, methods, conditions and the propagations in UserObject. All the purely virtual specifications in UserObject must be specified and implemented in ResourceA. As appears from Figure 7 the value of opState in ResourceA is derived from the attributes internalOpState and resourceBstate (which is inherited from UserObject), lines 46-49.

As has been mentioned earlier the implementation is essentially the same as when the objects belong to the same system. The difference is that a certain part of the functionality of the dependent object which uses ResourceB is implemented in UserObject. As earlier the code can automatically be generated from the specifications by a compiler. The declaration file which is generated from the specification of UserObject in Figure 12 is found in Figure 15.

The method checkConsistency in UserObject checks the dependence between the admState attributes, which are stated in the dependence scheme in Figure 13. The implementation of this method is shown in Figure 16. To check this condition the value of admState in UserObject must be read. This explains why there must be a purely virtual specification of admState in UserObject, even if this attribute is not really included in UserObject.

The declaration file which is generated by the specification of ResourceB in Figure 11 is found in Figure 17. The implementation of the method propagateOpstate is however somewhat different, compare Figure 8. The new value is propagated to the attribute ResourceB-state in the related userObject instance. The admState dependence between UserObject and ResourceB must be checked at updating of the object ResourceB as well as at updating of UserObject instances. Therefore, this condition is implemented in the method checkConsistency in ResourceB.

Figure 9 shows the declaration file, which is generated from the specification of ResourceA. The opState dependence of ResourceB is implemented by derivation of the value of opState from resourceBstate and internalOpState, compare Figure 20.

The advantages of the above described are the following.

Dependences and co-operation between uncoordinated objects can be specified and implemented in the same way as dependences between objects in the same subsystem. This implies that they are visible to and can be interpreted by a managing system.

The above illustrated process for maintenance of dependences between uncoordinated objects in a controlled and visible way facilitates implementation of a determinable management model in a layered architecture.

The degree of reuseability of the library components can be improved by a high degree of flexibility when combinations of components are concerned.

Claims

- 1. A method for implementing a managed object in a subsystem of a managed system in a management network with at least one managing system and at 5 least one managed system, for telecom or open systems, said managed system consisting of subsystems forming each a part of the managed system including one or more managed objects, characterized by implementing the managed objects in the subsystem, uncoordinatedly with respect to other subsystems, in such a way that they can be connected to and transmit messages to other objects in other subsystems, and without knowing the type of objects in the other subsystems, by designing a first object (330) for co-operation with an abstract object (332) defining an interface consisting of unimplemented methods, which may be called by the first object, and letting, at later design of a second object (334) unknown to said first object and intended to co-operate with the first object, the other object inherit the abstract object and implement the inherited methods, so that the first object at co-operation with the second object will consider it as being of said abstract type. 25
- A method according to claim 1, characterized by delegating, at call of a method defined in the interface of the abstract object, the call to the implementation in the real object by means of late binding.
- A method according to claims 1 or 2, characterized by reloading, in case of loading of said second object, said first object in the managed system by means of dynamic linking between the respective subsystems.
- A method according to any of claims 1-3, characterized by locating said first and second objects in first and second subsystems, respectively.
- A method according to claim 4, characterized by using dynamic linking to enable loading in the managed system of said second subsystem without reloading of said first subsystem.

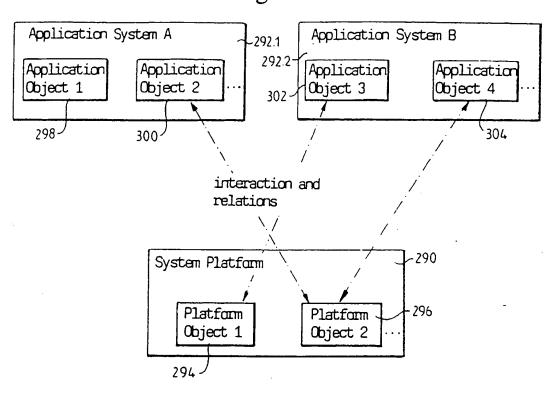
50

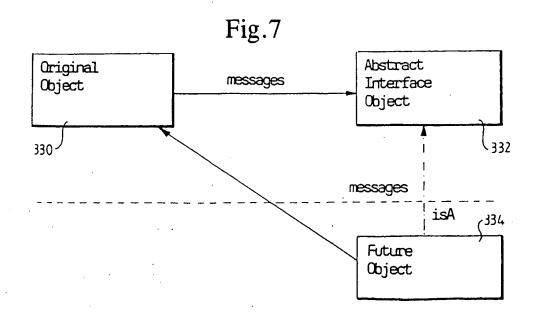
45

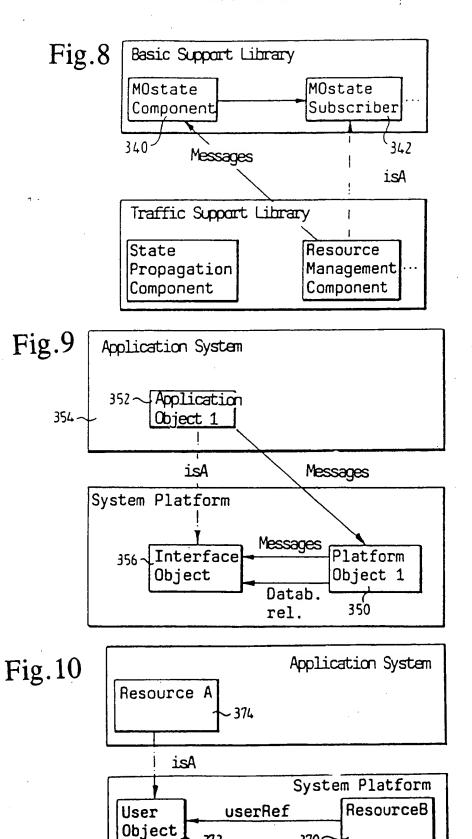
40

:22

Fig.6







-372

370-

```
OBJECT TYPE ResourceB IS
2
     ATTRIBUTES
       key : KeyType;
       admState : AdministrativeState;
       opState : OperationalState;
       userRef : REFERENCE TO UserObject
                  INVERSE OF Bref;
     PRIMARY KEY key;
     METHODS
       allocate() RETURNS Boolean;
10
11
12
     PRE-CONDITIONS
13
       deleteObject() ONLY IF
       admState = locked;
14
     PARTY TO ResourceAndUser;
15
16END:
```

```
25DEPENDENCY SCHEMA ResourceAndUser IS
    FOR ALL UserObject(a), ResourceB(b);
27
    RELATIONS a.Bref = b:
28
    POST-CONDITIONS
29
30
      NOT (a.admState = unlocked AND
31
            b.admState = locked)
32
      RULES
       WHEN COMMIT THEN CHECK CONSISTENCY;
33
34
35
    PROPAGATIONS
36
      WHEN b.opState = NewValue
37
      THEN CONCLUDE a.resourceBstate=NewValue;
38
39END;
```

```
400BJECT TYPE ResourceA IS
   BASE UserObject;
42
43
  ATTRIBUTES
44
      key : KeyType;
45
      admState : AdministrativeState;
      DERIVED opState = IF (InternalOpState=disabled OR
46
47
                             resourceBstate=disabled)
48
                         THEN disabled
49
                         ELSE enabled;
50
51
      internalOpState : OperationalState PRIVATE;
52
53
   PRIMARY KEY key;
54
55
   METHODS
56
      allocate() RETURNS Boolean;
57
58
   PRE-CONDITIONS
59
      deleteOblect() ONLY IF admState = locked;
160
61
   POST-CONDITIONS
62
      NOT (admState = unlocked AND Bref = NULL);
63
64END:
```

```
65 ifndef _UserObject hh
66 define _UserObject_hh_
67
68 include <PredefinedTypes.hh>
69 include "MOstateTypes.hh"
70.
71class ResourceB
73// OBJECT Type Userobject
75class Userobject
76(
77public
    static UserObject* open(Mode,const
    KeyType&, DbTransaction* transaction=
    NULL):
    virtual deleteObject() =0;
80
    virtual Boolean checkConsistency
81
    (ErrorMessage&);
82
    virtual AdministrativeState
83
    qetAdmState() =0;
    virtual void setAdmState
84
    (const AdministrativeState) ==;
    OperationalState getResourceBstate();
85
    setResourceBstate(const
86
    OperationalState);
    ResourceB* getBref();
87
    void setBref(ResourceB*);
88
89
90private:
91....
92....
93);
94
95 endif
```

```
97 include "UserObject.hh"
98 include "ResourceB.hh"
99
102// UserObject: METHOD checkConsistency
103//
104
105Boolean UserObject::checkConsistency(
          ErrorMessage& message)
106(
107
    Boolean flag = true;
108
    flag = !(getAdmState() unlocked &&
109
             getBref()->get admState()==locked);
110
111
    if (!flag) (
       createErrorMessage(1,message);
112
113
114
    return flag;
115)
116
```

```
ResourceB hh
117 ifndef
118 define ResourceB hh
119
120 include <PredefinedTypes.hh>
121 include "MOstateTypes.hh"
122
123class UserObject;
125// OBJECT TYPE ResourceB
127class ResourceB
128(
129public
    void deletobject();
130
131
    void deleteObject();
132
133 Boolean allocate();
134 virtual Boolean checkConsistency
     (ErrorMessage&);
135
136
     KeyType getKey();
     AdministrativeState getAdmState();
137
138
     void setAdmState (const
     AdministrativeState);
     OperationalState getOpState();
139
     void setOpState(const OperationalState);
140
     UserObject* getUserRef();
141
     void setUserRef(UserObject*);
142
143
144private
145
     void propagateOpState
     (const OperationalState);
     void opState (const OperationalState);
146
147
148
149);
150
151 endif
```

```
153 include "ResourceB.hh"
154 include "UserObject.hh"
155
157//
158// ResourceB: METHOD setOpState
159//
160
161void ResourceB: :setOpState(
          const OperationalState newValue)
162(
    OperationalState oldValue = getOpState();
163
164
165
    opState(newValue);
    if (newValue != oldValue)
166
167
       propagateOpState(newValue);
168
169
170)
171
173//
174// ResourceB: METHOD PropagateOpState
175//
176
177void ResourceB: :propagateOpState(
            const OperationalState newValue)
178(
    getUserRef()->resourceBstate(newValue)
179
180)
181
```

```
ResourceA hh
182 ifndef
183 define ResourceA hh
184
185 include <PredefinedTypes.hh>
186 include "UserObject.hh"
187
188// OBJECT TYPE ResourceA
19Oclass ResourceA : public UserObject
191(
192public
     static ResourceA* open(Mode,
193
194 const KeyType&, DbTransaction*
     transaction=NULL);
     void deleteObject ();
195
     virtual Boolean checkConsistency
196
     (ErrorMessage&);
197
198
     Boolean allocate();
199
200
     KeyType getKey();
201
     AdministrativeState getAdmState();
202
     void setAdmState (const
     AdministrativeState);
203
     OperationalState getOpState();
204
205private
     OperationalState getInternalOpState();
206
207
     void setInternalOpState (const
     OperationalState);
208
209
210):
211
212 endif
213
```

```
214 include "ResourceA.hh"
215.
216/////////////////////////////
217//
218// ResourceA: METHOD getOpState
219//
220
221OperationalState ResourceA: :getOpState()
222(
         (get InternalOpState() ==disabled
223
     if
          getResourceBstate() ==disabled)
224
225
      return disabled;
226
227
228
     else
229
230
      return enabled;
231
232)
233
```

(11) EP 0 817 422 A3

(12)

EUROPEAN PATENT APPLICATION

(88) Date of publication A3: 04.03.1998 Bulletin 1998/10

(51) Int. Cl.⁶: H04L 12/24

(43) Date of publication A2: 07.01.1998 Bulletin 1998/02

(21) Application number: 97117274.7

(22) Date of filing: 18.08.1993

(84) Designated Contracting States:

BE CH DE DK ES FR GB GR IE IT LI NL

(30) Priority: 28.08.1992 SE 9202488 05.02.1993 SE 9300363

(62) Document number(s) of the earlier application(s) in accordance with Art. 76 EPC: 93919758.8 / 0 627 143

(71) Applicant:
TELEFONAKTIEBOLAGET LM ERICSSON
126 25 Stockholm (SE)

(72) Inventors:

- Carebrand, Per-Arne
 118 42 Stockholm (SE)
- Svedberg, Johan
 115 24 Stockholm (SE)

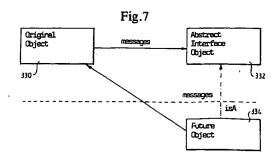
- Fantenberg, Johan
 112 47 Stockholm (SE)
- Talldal, Björn
 161 46 Bromma (SE)
- Palsson, Martin
 146 45 Tullinge (SE)
 Gilander Anders
- Gilander, Anders
 195 54 Märsta (SE)
- Sellstedt, Patrik
 124 71 Bandhagen (SE)
 Strömberg Stefan
- Strömberg Stefan
 141 52 Huddinge (SE)
- (74) Representative:
 Rosenquist, Per Olof et al
 Bergenstrahle & Lindvall AB,
 P.O. Box 17704
 118 93 Stockholm (SE)

(54) A method for implementing a managed object in a network management system

(57) The invention relates to a method for implementing a managed object in a subsystem of a managed system in a management network with at least one managing system and at least one managed system, for telecom or open systems, said managed system consisting of subsystems forming each a part of the managed system including one or more managed objects.

The managed objects are implemented in the subsystem, uncoordinatedly with respect to other subsystems, in such a way that they can be connected to and transmit messages to other objects in other subsystems, and without knowing the type of objects in the other subsystems. This is obtained by designing a first object (330) for co-operation with an abstract object (332) defining an interface consisting of unimplemented methods, which may be called by the first object, and letting, at later design of a second object (334) unknown to said first object and intended to co-operate with the first object, the other object inherit the abstract object and implement the inherited methods, so that the first object at co-operation with the second object will con-

sider it as being of said abstract type.



EP 0 817 422 A3



EUROPEAN SEARCH REPORT

Application Number
EP 97 11 7274.7
Page 1

	US 4782506 A (MAX) 1 November 1988 (* column 1, line 5 line 53, figures 1	IMILIAN SEVCIK), (01.11.88) 51 - column 3,	to claim	APPLICATION (Int. Cl.6) H04L 12/24
	1 November 1988 (* column 1, line 5	(01.11.88) 51 - column 3,	1-5	H04L 12/24
	* column 1, line 5	51 - column 3,		
		-,-	i	
				•
	EP 0442809 A2 (DIG	SITAL EQUIPMENT	1-5	
	CORPORATION), 21 A			
	(21.08.91)			
	* page 2, line 21	- line 29; page 3,		•
	line 28 - line 55,	, figure 1 *		
	•			
	US 4903263 A (RAJE	ENDRA PATEL ET AL),	1-5	
	20 February 1990	(20.02.90)		
	* column 1, line 8			
	column 12, line 49	- line 65, figure		
	-			TECHNICAL FIELDS
				SEARCHED (Int. Cl.6)
	•			H04L
				H04M H04Q
	•			
		•		,
	•			
1				
.				
1		<i>}</i>	,	
ŀ	•			
	The present search report has	been drawn up for all claims		
	Place of search	Date of completion of the search		Examiner
STOCK	HOTM	14 November 1997	ANDE	RS STRÖBECK
X : part Y : part doci A : tech	CATEGORY OF CITED DOCUM ticularly relevant if taken alone ticularly relevant if combined with ument of the same category notogical background -written disclosure	E: earlier pate after the fi	rinciple underlying the ent document, but put ling date cited in the application cited for other reason	olished on, or n

O FORM 1503 03.82 (I

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ BLACK BORDERS
IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
FADED TEXT OR DRAWING
BLURRED OR ILLEGIBLE TEXT OR DRAWING
☐ SKEWED/SLANTED IMAGES
☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
☐ GRAY SCALE DOCUMENTS
☐ LINES OR MARKS ON ORIGINAL DOCUMENT
☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
□ OTHER:

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.